Automatic Pitch-Synchronous Phonetic Segmentation

Jindřich Matoušek, Jan Romportl

Dept. of Cybernetics, Faculty of Applied Sciences, University of West Bohemia, Czech Republic
jmatouse@kky.zcu.cz, rompi@kky.zcu.cz

Abstract

This paper deals with an HMM-based automatic phonetic segmentation (APS) system and proposes to increase its performance by employing a pitch-synchronous (PS) coding scheme. Such a coding scheme uses different frames of speech throughout voiced and unvoiced speech regions and enables thus better modelling of each individual phone. The PS coding scheme is shown to outperform the traditionally utilised pitch-asynchronous (PA) coding scheme for two corpora of Czech speech (one female and one male) both in the case of a base (not-refined) APS and in the case of a CART-refined APS. Better results were observed for each of the voicing-dependent boundary types (unvoiced-unvoiced, unvoiced-voiced, voiced-unvoiced and voiced-voiced).

Index Terms: automatic phonetic segmentation, pitch-synchronous coding, hidden Markov models, speech synthesis, unit selection

1. Introduction

Automatic phonetic segmentation (APS) is a process of detecting boundaries between phones in speech signals. Since manual segmentation is labour-intensive and time-consuming, the automation of the process is very important especially when many speech signals are to be segmented. This is exactly the case of *unit selection*, a very popular and still the most prevalent text-to-speech (TTS) synthesis technique. Being a corpus-based concatenative speech synthesis method, the principle of unit selection is to concatenate pre-recorded speech segments (extracted from natural utterances using the automatically segmented boundaries) carefully selected from a large speech corpus according to phonetic and prosodic criteria imposed by the synthesised utterance. It is evident that automatic phonetic segmentation affects the quality of synthetic speech produced by a unit-selection-based TTS system.

The most often used approaches to automatic phonetic segmentation are based on *hidden Markov models* (HMMs), a statistical framework widely used in the area of automatic speech recognition. The idea in APS is to apply similar procedures as for speech recognition. However, instead of the recognition, so-called *forced-alignment* is performed to find the best alignment between HMMs and the corresponding speech data, producing a set of boundaries which delimit speech segments belonging to each HMM. Briefly, each phone unit is modelled by a context-dependent HMM (CD-HMM) or context-independent HMM (CI-HMM). Firstly, the model parameters are trained on the basis of a collection of speech data (described by *feature vectors*) with the corresponding phonetic transcripts. Typically, the *embedded training* strategy is employed in which a sequence of models associated with the given phonetic transcript are concatenated and all model parameters are simultaneously updated through the Baum-Welch algorithm [1, 2]. When some manually segmented data are available, the so-called *isolated-unit training* utilising the Baum-Welch algorithm with model boundaries fixed to manually segmented ones can be also employed. Secondly, the trained HMMs are employed to align a speech signal along the associated phonetic transcript by means of Viterbi decoding. A simplified scheme of an HMM-based APS system is given in Fig. 1.

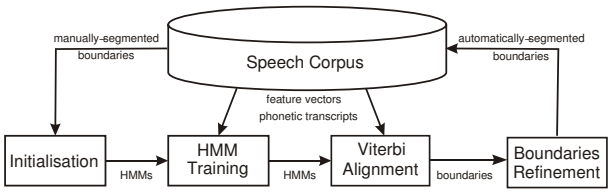

Figure 1: *Simplified scheme of HMM-based automatic phonetic segmentation.*

As the performance of an HMM-based APS system (denoted as *base APS* henceforth) is usually not accurate enough to be directly applied to TTS, various modifications and post-processing techniques have been developed. These techniques usually increase the segmentation accuracy by refining the initial segmentations from a base APS system. Various methods are utilised for the refinement. Some of them try to fix boundary-specific discrepancies between automatic and manual segmentations by means of statistically motivated approaches like classification and regression trees (CART), neural networks or support vector machines [3, 4, 5, 6, 7]. Other studies propose to refine the boundaries by employing an explicit (local) boundary model with the use of various acoustic features [4, 8, 9, 10]. Some authors also propose to modify the underlying Baum-Welch algorithm [11] or to employ minimum boundary error (MBE) criterion instead of traditionally used maximum-likelihood (ML) criterion [7]. Recently, multiple APS system framework was proposed where several parallel APS systems (base APS systems with different configurations and/or various post-processing techniques) are employed to segment the same data and the final segmentation results are then obtained as a combination of the results from each APS [12, 2, 13].

Although much work has been done, there are still some shortcomings in the HMM-based APS. As feature vectors used to train HMMs are usually extracted with a given step (typically 5-10 ms), the accuracy of boundary detection is therefore limited. Since there is an effort to concatenate units in a consistent way in TTS systems, the automatically detected boundaries are often moved to some distinctive points in speech signals (usually points of the most/least rapid spectral change or points

Support for this work was provided by the Ministry of Education of the Czech Republic, projects No. 2C06020 and No. MSMT LC536. The access to the METACentrum clusters provided under the research intent MSM6383917201 is highly appreciated.

of the principal excitation of vocal tract, so-called *pitch-marks*) which could possibly introduce certain bias to the APS results. This paper focuses on the base APS system and proposes to increase its performance by employing a *pitch-synchronous coding scheme*. As the boundaries detected by such an automatic pitch-synchronous phonetic segmentation system are implicitly placed on pitch-marks, there is no need to move the boundaries any more.

The paper is organised as follows. The concept of a pitch-synchronous coding scheme is introduced in Section 2. In Section 3, corpora used in our experiments are presented. Experiments with different coding schemes and the results of the performance evaluation and their discussion are provided in Sections 4 and 5. Finally, conclusions are drawn in Section 6.

2. Pitch-synchronous coding scheme

Traditionally, the *pitch-asynchronous* (PA) coding scheme is employed for modelling speech. In this scheme, a uniform analysis frame of a given length l_{u} is defined and slid along the whole speech signal of an utterance with a fixed shift s_{u}. The length is usually set to comprise frequency characteristics of the speaker ($l_{\mathrm{u}} \approx 2T_0$ where T_0 is a maximum pitch period of the speaker). The shift is usually set to 5-10 ms which roughly corresponds to T_0. The accuracy of such a scheme is questionable mainly in unvoiced speech regions where both frequency and time resolutions are not accurate, especially for dynamic sounds like plosives. Even in voiced speech regions, the PA scheme due to the changes in fundamental frequency (F_0) does not extract frames in a consistent way.

In order to extract the frames for coding in a *pitch-synchronous* way, our pitch-mark detection algorithm described in [14] was employed. By the term *pitch-marks* we mean the locations of principal excitation of vocal tract (corresponding to glottal closure instants) in speech signals. The idea here is that, knowing these locations, approx. two-pitch-period-sized frames of speech centred on each pitch-mark can be efficiently and consistently extracted from voiced speech. The pitch-mark detection algorithm works in multiple phases and utilises both glottal and speech signals. In the 1st phase, the glottal signal is used for the precise estimation of F_0 contour of the utterance. Next, pitch-mark candidates are generated on the basis of both glottal and speech signals. In the 3rd phase, the best sequence of pitch-marks is found in the set of the candidates by means of dynamic programming. Finally, the selected pitch-mark sequence can be a subject of post-processing in which errors with "doubling" and "halving" F_0 are fixed. The overall accuracy of the pitch-mark detection algorithm is approx. 98% [14].

In a general *pitch-synchronous* (PS) coding scheme, each frame to be extracted for coding is defined both by its position in a speech signal and its length. So, in general, a sequence of different frames must be given in order to perform pitch-synchronous coding. In voiced speech, let us denote the position of the frame $f^{(i)}$ as $p_{\mathrm{v}}^{(i)}$ and its length as $l_{\mathrm{v}}^{(i)}$ ($i = 1, \ldots, N$, where N is the number of all frames in the utterance). In our case, the positions are given by the detected pitch-marks and taken as the central positions of the frames, i.e. each "voiced frame" $f^{(i)}$ is defined as $\langle p_{\mathrm{v}}^{(i)} - \frac{l_{\mathrm{v}}^{(i)}}{2}, p_{\mathrm{v}}^{(i)} + \frac{l_{\mathrm{v}}^{(i)}}{2} \rangle$. The lengths of the particular frames generally vary according to the instantaneous pitch period $T_0^{(i)}$ (a reasonable value is $2T_0^{(i)}$), but can be also fixed to a given value. In unvoiced speech, no pitch-marks are defined because there is no activity of vocal cords during unvoiced speech regions. Therefore, standard PA coding scheme with a fixed frame length l_{u} and a fixed frame shift s_{u} is employed here. To be compatible with the voiced regions of speech, each "unvoiced frame" $f^{(i)}$ can be defined as $\langle p_{\mathrm{u}}^{(i)} - \frac{l_{\mathrm{u}}}{2}, p_{\mathrm{u}}^{(i)} + \frac{l_{\mathrm{u}}}{2} \rangle$ where $p_{\mathrm{u}}^{(i)}$ are the central positions of the unvoiced frames with the distance s_{u} between them. At the boundaries between the unvoiced and voiced speech regions, the unvoiced frames are placed in pitch-asynchronous manner until the beginning of an unvoiced frame oversteps the beginning of the first voiced frame. At the boundaries between the voiced and unvoiced regions, the first unvoiced frame is centred on the end of the last voiced frame. As a result, a sequence of frames $f^{(i)}$ ($i = 1, \ldots, N$) consisting of the subsequences of both voiced and unvoiced frames is available for coding. The illustration of the pitch-synchronous coding scheme is given in Fig. 2.

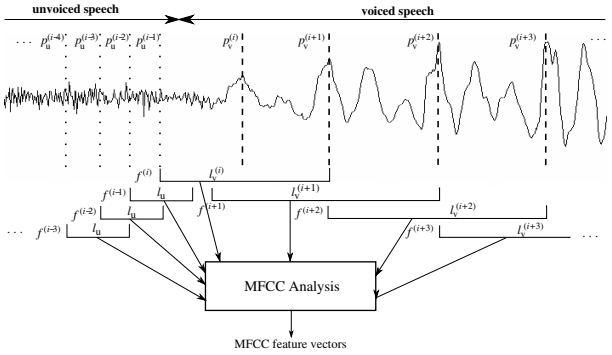

Figure 2: *Illustration of a pitch-synchronous coding scheme.*

3. Description of data

For our experiments we used two Czech phonetically and prosodically rich speech corpora, one of a female voice (FC) and one of a male voice (MC). The utterances included in the corpora were carefully selected, spoken by a professional speaker in an anechoic chamber, recorded at 16-bit precision with 48 kHz sampling frequency (later down-sampled to 16 kHz) and carefully annotated both on the orthographic and phonetic level [15]. Phonetic transcripts for all utterances plus some manual segmentations from a phonetic expert were available. In order to train the APS systems, a feature vector was computed for each frame according to the pitch-synchronous coding scheme described in Section 2 using 12 mel-frequency cepstral coefficients (MFCCs), log energy and their delta and delta-delta coefficients (39 coefficients for each frame in total).

The FC speech corpus consists of 5,139 utterances (6.70 hours of speech excluding the leading and trailing pauses, 300,969 phone boundaries in total). 58 utterances were segmented manually (9.94 minutes, 7,618 phone boundaries in total), 46 manually segmented utterances were used to initialise APS systems and 12 were used for testing. The MC corpus consists of 12,242 utterances (17.69 hours of speech excluding the pauses, 675,809 phone boundaries in total), 90 of them were segmented manually (11.71 minutes, 7,789 phone boundaries in total). 70 manually segmented utterances were used to initialise APS systems and 20 manually segmented utterances were used for testing. In order to reduce the labour-intensive and time-consuming manual segmentation, the amount of the manually segmented data was intentionally kept to minimum for both corpora.

4. Experiments & Results

All experiments with the automatic phonetic segmentation were carried out following the scheme shown in Fig. 1 and using the HTK software [1]. Only experiments with different pitch-synchronous coding schemes were conducted – all other components of the APS system were fixed according to our previous experiments: each HMM topology was fixed as 3-state left-to-right without any state skipping (with the exception of the pause models) with each state modelled using a single Gaussian mixture, the usage of both CI-HMMs and CD-HMMs, the employ of both isolated (for initialization) and embedded (for re-estimation) unit training procedures. Such a setting was found to yield the best segmentation results in our research [3, 16], although some other studies (e.g. [4, 2]) reported that other configurations (and especially the use of CI-HMMs with more mixture components per state) could lead to better results. The reason for using the aforementioned configuration in our experiments could be seen in having a relatively small number of manually segmented utterances available. This was partially confirmed in [2], where, on the other hand, an enormous number of 2,000 manually segmented utterances were available.

Basically, three different configurations of the PS coding scheme described in Section 2 were researched (for the sake of simplification, PS$\{l/s\}$ denotes here PS coding scheme with l being the frame length and s frame shift in voiced regions; the symbol '$\bullet$' stands for pitch-synchronous frame length, or shift respectively):

PS$\{\bullet/\bullet\}$ In this pure PS coding scheme, each voiced frame $f^{(i)}$ is centred around the corresponding pitch-mark $p_{\mathrm{v}}^{(i)}$ exactly as mentioned in Section 2. To ensure symmetrical centering around the pitch-mark, the length of the frame is computed as

$$l_{\mathrm{v}}^{(i)} = 2 \cdot \max\left\{p_{\mathrm{v}}^{(i)} - p_{\mathrm{v}}^{(i-1)}, p_{\mathrm{v}}^{(i+1)} - p_{\mathrm{v}}^{(i)}\right\}. \quad (1)$$

The length of the first frame in a voiced speech region is computed as $2 \cdot (p_{\mathrm{v}}^{(i+1)} - p_{\mathrm{v}}^{(i)})$, and similarly, the length of the last frame is computed as $2 \cdot (p_{\mathrm{v}}^{(i)} - p_{\mathrm{v}}^{(i-1)})$.

PS$\{l_{\mathrm{v}}/\bullet\}$ Again, each voiced frame is centred around the corresponding pitch-mark, but the length of all voiced frames was set to a fixed value ($l_{\mathrm{v}} = 20$ ms for FC corpus and $l_{\mathrm{v}} = 25$ ms for MC, respectively).

PS$\{l_{\mathrm{v}}/s_{\mathrm{v}}\}$ Here, a PA coding scheme was employed in voiced speech regions (different from the scheme in unvoiced speech regions), independently on pitch-marks. The idea for this configuration was that no pitch-mark detection (and possibly also no glottal signal recording) would be required, and only information about voicing would be needed. The same frame lengths as in PS$\{l_{\mathrm{v}}/\bullet\}$ and shift $s_{\mathrm{v}} = 10$ ms were used for both speech corpora.

In the unvoiced speech regions, the identical setting of the PA coding scheme ($l_{\mathrm{u}} = 6$ ms and $s_{\mathrm{u}} = 3$ ms) for all three configurations described above and for both corpora was employed based on our previous experiments.

The results of our experiments are shown in Table 1. The standard PA coding scheme PA$\{l_{\mathrm{u}}/s_{\mathrm{u}}\}$ was used as the baseline ($s_{\mathrm{u}} = 10$ ms for both corpora, because for $s_{\mathrm{u}} < 10$ short cross-word pauses tended to be missed during the alignment which resulted in gross segmentation errors). For performance evaluation, the mean absolute error (MAE) and percentage of boundaries deviating less than the given tolerance time region from the manually determined boundaries are often utilised. As the manual segmentation is an error-prone process, relatively high tolerance regions like 20 ms or the mean value over more tolerance regions (MT) are often used to get more robust results. In our experiments, MAE, tolerance regions of 10, 20 and 50 ms and MT computed from tolerance regions of 5, 10, 20, 30 and 50 ms are shown.

Table 1: *Segmentation results for base APS systems.*

coding scheme	MAE (ms)	10ms (%)	20ms (%)	50ms (%)	MT (%)	corpus
PA$\{20/10\}$	9.32	68.89	89.47	98.43	78.16	FC
PS$\{20/10\}$	8.41	73.12	90.12	99.39	80.73	
PS$\{20/\bullet\}$	8.16	75.75	91.56	99.28	81.95	
PS$\{\bullet/\bullet\}$	**6.94**	**77.59**	**92.05**	**99.40**	**83.83**	
PA$\{25/10\}$	8.86	70.42	90.10	99.35	78.30	MC
PS$\{25/10\}$	9.63	59.72	91.45	99.42	76.20	
PS$\{25/\bullet\}$	10.25	63.82	88.09	99.48	75.86	
PS$\{\bullet/\bullet\}$	**8.11**	**73.38**	**92.42**	**99.61**	**80.44**	

5. Discussion

As can be seen in Table 1, the pure pitch-synchronous scheme PS$\{\bullet/\bullet\}$ yields the best results for both corpora. For the female speech corpus (FC), all PS coding schemes outperform the baseline PA scheme in terms of all performance indexes. This is also true for the scheme PS$\{25/10\}$, in which no pitch-marking was performed and the different coding was applied based on the voicing detection only.

The analysis of the segmentation results for the male speech corpus (MC) is a bit complicated. Again, the best results were obtained for the pure PS coding scheme in all performance indexes, but the other PS schemes outperform the baseline PA scheme rather when higher tolerance regions (20 and 50 ms) are considered. The absolute results for FC and MC also differ. Substantially better results in terms of MAE and 10-ms tolerance region were obtained for FC. On the other hand, the better performance in terms of the higher tolerance regions was reached for MC. The explanation of these findings is still under consideration. Preliminary, we believe that the ambiguous results could root in the manually segmented data. Following the performance evaluation, there are more segmentation errors of lower relevance in the MC corpus which could indicate that there are some inconsistencies in the manually segmented data.

In Table 2, segmentation results for particular boundaries with respect to the voicing nature of both boundary phones are shown. Baseline PA schemes (PA$\{20/10\}$ for FC and PA$\{25/10\}$ for MC – both denoted here as PA) and the PS schemes with the best results from Table 1 (denoted here as PS) are compared. As can be seen, PS coding schemes yielded better results for all boundary types.

As CD-HMMs are generally known to introduce boundary-dependent biases in the segmentation results [3, 12, 2] (probably caused by training the same CD-HMM with phones of the same specific context [4]), we also applied a post-processing technique to remove the biases. Unlike [3], a more general *classification and regression tree* (CART) technique was utilised to compute the biases [5, 2]. In our approach, robust bias estimates are obtained by traversing the tree with respect to the phonetic features of phones adjacent to the boundary. The tree is built by clustering the deviation between manually and automatically segmented boundaries respecting the "phonetic type"

Table 2: *Segmentation results for unvoiced-unvoiced (U-U), unvoiced-voiced (U-V), voiced-unvoiced (V-U) and voiced-voiced (V-V) boundaries.*

bound. type	MAE (ms)		<20ms (%)		coding scheme
U-U	12.17	15.11	85.71	83.44	PA
	9.43	**13.89**	**87.50**	**85.62**	**PS**
U-V	6.62	9.52	95.17	90.99	PA
	4.11	**8.14**	**95.24**	**96.39**	**PS**
V-U	9.49	8.37	91.61	90.72	PA
	6.42	**7.88**	**94.84**	**92.81**	**PS**
V-V	10.16	9.18	88.73	89.35	PA
	7.75	**8.73**	**88.91**	**90.15**	**PS**
corpus	FC	MC	FC	MC	

Table 3: *Comparison of segmentation results of base APS systems (the 1st row in each box) and CART-refined APS systems (the 2nd row in each box).*

coding scheme	MAE (ms)	10ms (%)	20ms (%)	50ms (%)	MT (%)	corpus
PA{20/10}	9.32	68.89	89.47	98.43	78.16	FC
	6.65	80.15	95.64	99.39	84.14	
PS{•/•}	6.94	77.59	92.05	99.40	83.83	
	5.55	**83.73**	**96.39**	**99.64**	**86.96**	
PA{25/10}	8.86	70.42	90.10	99.35	78.30	MC
	5.75	83.95	96.38	99.87	85.76	
PS{•/•}	8.11	73.38	92.42	99.61	80.44	
	5.53	**85.17**	**96.63**	**99.61**	**86.96**	

of the boundaries. In order to reduce the possibility of misleading the clustering procedure by a small number of gross errors (e.g. caused by imperfect manual segmentation or incorrect phonetic transcripts of the training data), the deviation x was confined within the region $[-1, 1]$ applying the sigmoid function

$$f(x) = \frac{2}{1 + \exp(-\beta x)} - 1 \tag{2}$$

where the slope parameter $\beta = 0.08\ \mathrm{ms}^{-1}$ as proposed in [2]. To refine the automatic segmentation by removing the biases b, the clustered sigmoid-transformed bias b_{cl} had to be transformed to the original domain by applying the inverse transform

$$b = \frac{1}{\beta} \log \frac{1 + b_{\mathrm{cl}}}{1 - b_{\mathrm{cl}}}. \tag{3}$$

EST tool *wagon* [17] was used in these experiments. The segmentation results after the CART-based post-processing are shown in Table 3. As can be seen, CART-based refinement considerably improved the segmentation results, especially for MC corpus. Moreover, the refinement did not affect the superiority of the PS coding scheme over the PA scheme. Hence, utilising the proposed PS segmentation scheme together with a post-processing refinement technique should yield better results than when refining the standard PA segmentation scheme.

6. Conclusions

In this paper, the use of the pitch-synchronous coding scheme within the HMM-based APS system was researched. Having compared the influence of the pitch-synchronous and the standard pitch-asynchronous schemes on the segmentation results, the proposed PS scheme was shown to yield better results both for the base APS system and the CART-refined system. The performance was better for all voicing-dependent boundary types. This encourages us to claim that the proposed PS APS together with a post-processing refinement technique should yield better results than when refining the standard PA APS. Moreover, we believe that utilising the pitch-synchronicity as another aspect in the multiple APS system framework [2] could further improve the segmentation accuracy of such a system.

7. References

[1] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*. Cambridge, U.K.: Cambridge University, 2006.

[2] S. S. Park and N. S. Kim, "On using multiple models for automatic speech segmentation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2202–2212, 2007.

[3] J. Matoušek, D. Tihelka, and J. Psutka, "Automatic segmentation for Czech concatenative speech synthesis using statistical approach with boundary-specific correction," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 301–304.

[4] D. Toledano, L. Gómez, and L. Grande, "Automatic phonetic segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 617–625, 2003.

[5] J. Adell and A. Bonafonte, "Towards phone segmentation for concatenative speech synthesis," in *Proc. Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 139–144.

[6] K.-S. Lee, "MLP-based phone boundary refining for a TTS database," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 3, pp. 981–989, 2006.

[7] J.-W. Kuo, H.-Y. Lo, and H.-M. Wang, "Improved HMM/SVM methods for automatic phoneme segmentation," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 2057–2060.

[8] S. Boonsuk, P. Punyabukkana, and A. Suchato, "Phone boundary detection using selective refinements and context-dependent acoustic features," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 1362–1365.

[9] A. Sethy and S. Narayanan, "Refined speech segmentation for concatenative speech synthesis," in *Proc. Int. Conf. on Spoken Language Processing*, Denver, USA, 2002, pp. 149–152.

[10] Y.-J. Kim and A. Conkie, "Automatic segmentation combining an HMM-based approach and spectral boundary correction," in *Proc. Int. Conf. on Spoken Language Processing*, Denver, USA, 2002, pp. 145–148.

[11] D. Huggins-Daines and A. Rudnicky, "A constrained Baum-Welch algorithm for improved phoneme segmentation and efficient training," in *Proc. Interspeech*, Pittsburgh, USA, 2006, pp. 1205–1208.

[12] J. Kominek and A. Black, "A family-of-models approach to HMM-based segmentation for unit selection speech synthesis," in *Proc. Interspeech*, Jeju Island, Korea, 2004, pp. 1385–1388.

[13] S. Jarifi, D. Pastor, and O. Rose, "A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis," *Speech Communication*, vol. 50, pp. 67–80, 2008.

[14] M. Legát, J. Matoušek, and D. Tihelka, "A robust multi-phase pitch-mark detection algorithm," in *Proc. Interspeech*, Antwerp, Belgium, 2007, pp. 1641–1644.

[15] J. Matoušek, D. Tihelka, and J. Romportl, "Building of a speech corpus optimised for unit selection TTS synthesis," in *Proc. Int. Conf. on Lang. Resources and Evaluation*, Marrakech, Morocco, 2008.

[16] J. Matoušek, D. Tihelka, and J. Psutka, "Experiments with automatic segmentation for Czech speech synthesis," *Lecture Notes in Computer Science*, vol. 2607, pp. 287–294, 2003.

[17] P. Taylor, R. Caley, A. Black, and S. King, "Edinburgh speech tools library: System documentation," http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0/, 1999.