Capturing Speaker-Independent Prosodic Information by Syntax Tree-Based Prosody Modelling

梁/*LIANG* 晖/*Hui, Sarah Hoffmann*

Speech Processing Group, Computer Engineering & Networks Lab, ETH Zürich, Switzerland

{liangh,hoffmann}@tik.ee.ethz.ch

Abstract

A syntax tree-based prosody generation module had been previously developed for our own speaker-dependent, concatenative speech synthesiser SVOX. In this technical report, we mainly examined the possibility of this syntax tree-based module capturing speaker-independent prosodic information. Apart from that, we also looked at the possibility of supplementing synthetic prosodic features with speaker-dependent cues, using this syntax tree-based module and a small amount of "adaptation data". We experimented on the German language. Speech samples presented to listeners, who speak German as their first language, were generated by an HMM-based speech synthesiser whose conventional prosody prediction mechanism was replaced with this syntax tree-based prosody generation module. Subjective evaluations suggested a layer of this syntax tree-based prosody generation module could capture speaker-independent prosodic information of German and that given a small amount of "adaptation data" another layer of this module could reproduce basic global speaker-dependent prosodic cues.

Index Terms: speaker-independent prosodic information, syntax tree, prosody generation

1. Introduction

The SIWIS project [1], *S*poken *I*nteraction *w*ith *I*nterpretation in *S*witzerland, is a continuation of the EMIME project [2, 3] and aims for a more personalised speech-to-speech translation system: In addition to transferring the voice characteristics of a user (i.e. speaker-specific spectral characteristics) into output synthesised speech, the SIWIS project is also aimed to reproduce the user's personal prosodic characteristics in synthesised spoken translations. Both aims would require *the separation of language-related information from speaker-related information*. The research conducted in this technical report is towards the separation for the second aim.

The HMM-based speech synthesis framework [4, 5, 6] is infinitely preferable to SIWIS, as it has proved highly adaptable [7, 8, 9]. Speaker and language factorisation was demonstrated to be a solution to separating language-related information from speaker-related information in the HMM-based speech synthesis framework, and was applied to both spectral and prosodic features [10]. Nonetheless, the conventional prosody prediction mechanism itself is a weakness of HMM-based speech synthesis, because it is difficult for HMM states to capture long-term tendencies of variation of prosodic features. Hence, different prosody modelling techniques for speech synthesis that have the potential of separating language-related information from speaker-related information would be of more interest to us.

Unfortunately, it appears that there has been little other research conducted on separating language-related prosodic information from speaker-related prosodic information for speech synthesis. Previously a high-quality concatenative speech synthesiser called SVOX [11] was developed in our laboratory. The most recent accomplishment [12] for SVOX is a syntax tree-based prosody generation module that models prosody with two stages: one models the cumulative effects ranging from the sentence level to the word level, and the other models phone-level effects. Apparently this 2-stage module could have the potential of capturing language-related and speaker-related prosodic information separately, although it has been developed only on speech data from a single speaker thus far.

In this technical report we mainly examine the possibility of our syntax tree-based prosody generation module capturing speaker-independent prosodic information. In addition, we also look at the possibility of supplementing synthetic

TIK-Report No. 352, ETH Zürich, June 2014

prosodic features with speaker-specific cues, using this syntax tree-based module and a small amount of speech data from target speakers. First of all, we give a short introduction to our syntax tree-based prosody generation module.

2. Syntax Tree-Based Prosody Generation

A prosody generation module for concatenative speech synthesis that makes direct use of syntax trees to predict duration and pitch had been developed in our laboratory. Its technical details can be found in [12]. In brief, two stages are involved in this prosody generation module: (i) a prosody contour at the word level is generated from the syntax tree of a sentence (see sections 2.1, 2.2 and 2.3); (ii) the prosody contour, together with contextual information about phones in the sentence, is further processed to yield concrete prosodic features (F_0 and duration) for this sentence (see section 2.4).

The input for training the entire prosody generation module is concrete F_0 curves and forced-alignment results at the phone level. Please note that this prosody generation module employs a cepstrogram-based F_0 extraction algorithm that produces continuous F_0 curves [13], so that voice and unvoiced regions can be treated in the same fashion.

2.1. Prosody contours

Effectively, the above-mentioned word-level prosody contour is merely a description of concrete prosodic features, and is portrayed as a sequence of vectors. Each of these word-level vectors consists of seven elements and corresponds to a single word in the sentence. The seven elements associated with a word are:

- normalised F_0 mean: $\dfrac{F_{0,\text{word}}^{\text{mean}} - F_{0,\text{corpus}}^{\text{min}}}{F_{0,\text{corpus}}^{\text{max}} - F_{0,\text{corpus}}^{\text{min}}} \in [0,1]$
- degree of flatness of the F_0 curve of the word: mean squared error between the F_0 curve and its optimal linear estimate in the sense of least squares
- gradient of the F_0 curve of the word: the gradient of the optimal linear estimate in the sense of least squares for the F_0 curve
- length of the pause before the word (0 = no pause)
- length of the pause after the word (0 = no pause)
- normalised duration of the word: $\dfrac{\sum_{p=1}^{P} d_p}{\sum_{p=1}^{P} d_{p,\text{corpus}}^{\text{mean}}}$, where d_p means the duration of the p-th phone in the word, and $d_{p,\text{corpus}}^{\text{mean}}$ means the average duration of this phone in the entire corpus
- gradient of duration of the word: the gradient of the optimal linear estimate in the sense of least squares for the set of points $\left\{ \left(p, \dfrac{d_p}{d_{p,\text{corpus}}^{\text{mean}}} \right) \middle| p = 1, 2, \cdots, P \right\}$

2.2. Deriving prosody contours from syntax trees

The syntax tree of a sentence is generated by the SVOX syntax analysis [11], which is based on a chart-parser using sentence-level and word-level definite clause grammars and a lexicon containing orthographic and phonetic transcriptions. In addition, the syntax tree is simplified by stripping any additional attributes, so that all the nodes on the tree are described only with constituents of underlying grammar rules. Every leaf node corresponds to a word, with a phonetic transcription and lexical stress information retained as well. Figure 1 shows an example. Syntax trees of training sentences are treated as invariable in subsequent training steps.

Each node on a syntax tree is considered a potential prosodic unit that may contribute to the word-level prosody contour of the sentence. To that end, each node is assigned a *mutator function*, which describes the local contribution of the node to the 7-element vector of every word on the subtree of this node. Consequently, the 7-element vector of a word is the superpositional result of all the mutator functions along the path from the root node to the corresponding leaf (e.g. the path in bold type that ends at "saw" in Figure 1). Then these vectors constitute the prosody contour of the sentence.

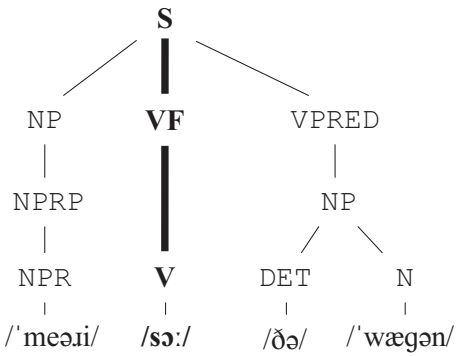

Figure 1: *simplified syntax tree* [12]

2.3. Mutator functions

The mutator function assigned to a node is effectively a *classification and regression tree* (CART), which describes the relationship between the 7-element vectors associated with leaves on the subtree of this node and the following inputs:

- constituent of the parent node
- constituents of the adjacent left and right siblings
- constituents of other siblings
- punctuation before and after the node
- number of syllables on the subtree of the node
- $\star$ output of the mutator function of the parent node

The way the grammar is structured ensures that the set of relevant parent or sibling constituents is much smaller than the full set of constituents used in the grammar. The feature space is greatly reduced and therefore we estimate *one* mutator function *per* constituent (i.e. one CART per constituent). All the CARTs are trained with standard algorithms [14]. CARTs at a higher level of a syntax tree are trained and fixed, and then those below them get trained.

2.4. Generation of concrete prosodic features

Concrete prosodic features are generated by artificial neural network (ANN), a built-in module of SVOX. The ANN for duration has exactly one output that describes the length of a phone. The ANN for F_0 outputs a sequence of normalised F_0 values that form the concrete F_0 curve of a syllable. The normalised F_0 values are between 0 and 1, and by default this range is mapped to the F_0 range of a target speaker in the end. The input to the ANNs is contextual information about phones, which is similar to that employed in HMM-based speech synthesis, such as:

- type of the current phone (fricative, vowel, etc)
- length of the syllable containing the phone
- whether the syllable containing the phone is stressed
- position of the syllable containing the phone in the current word

as well as corresponding word-level prosody contours as defined in section 2.1.

3. Research Plan

Considering the fact that our syntax tree-based module does not produce any spectral features, we plan to synthesise speech by replacing the prosody prediction mechanism of HMM-based speech synthesis with this syntax tree-based module, whilst in the training stage, a conventional HMM-based speech synthesiser and the syntax tree-based prosody generation module are trained separately. The SVOX syntax analysis module is employed as a common TTS front-end. Figure 2 shows how HMM-based speech synthesis is combined with our syntax tree-based prosody generation module in the synthesis stage.

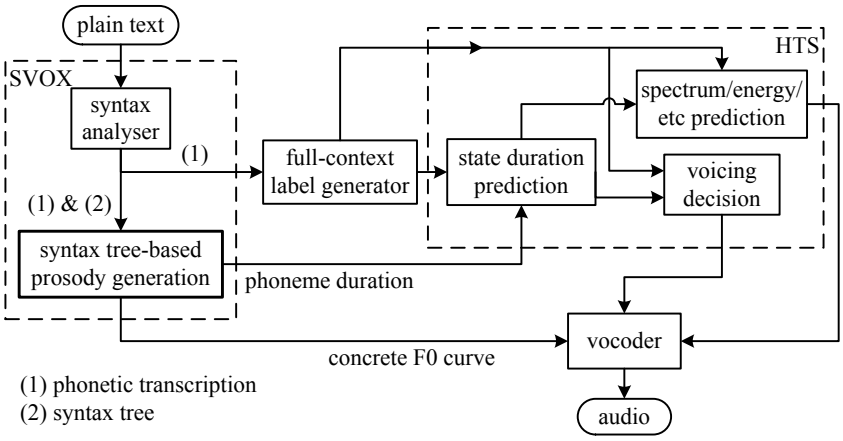

Figure 2: *hybrid framework employed in the synthesis stage*

Since the syntax tree-based prosody generation module was previously developed only on speech data from a single speaker, we are much interested in examining the possibility of this module capturing speaker-independent prosodic information. As explained in section 2, two stages are involved in this prosody generation module. The first stage is to generate a prosody contour at the word level, using mutator functions at the same and higher levels. In the second stage, the word-level prosody contour is taken as a guide for the generation of concrete prosodic features, which are at a lower level. This fact implies the first stage could be primarily handling the general tendency of prosody variation of a sentence and that the second stage could be largely handling more specific, local patterns of prosody. In other words, *mutator functions* trained in the first stage might capture more speaker-independent prosodic information, and *ANNs* trained in the second stage might capture more speaker-dependent prosodic cues. Consequently, we intend to find out the extent of speaker-independence of mutator functions by checking how applicable they are to ANNs trained on speech data from a different speaker.

4. Experiments

Speech samples generated for the following experiments have been assembled on a demo webpage (http://goo.gl/qyn46F) for the purpose of better presenting our experimental results and analysis.

4.1. Speech Data and Systems

4.1.1. HMM-based speech synthesiser

We trained a conventional average-voice HMM-based speech synthesiser on the German corpora PhonDat1 and PhonDat2 [15] (217 speakers, 18972 utterances/20.6 hours in total) using the HTS-2010 system [16]. The HMM topology was five-state, single Gaussian-per-state and left-to-right with no skip.

This average-voice synthesiser was then adapted by CSMAPLR [9] to the voices of two female (GF2 & GF3) and two male (GM3 & GM7) target German speakers, which were chosen from the EMIME bilingual corpus [17]. 75 utterances in German in each speaker's voice were used as adaptation data.

Speech features of the above-mentioned data were 39th-order STRAIGHT [18] mel-cepstra plus one dimension of en-

ergy, $\log F_0$, 21-dimensional band aperiodicity, and their delta and delta-delta coefficients, extracted from 16kHz recordings with a window shift of 5ms.

This adapted HMM-based speech synthesiser was regarded as *system D* in the following assessments and generated non-prosodic features for the output of our syntax tree-based prosody generation module.

4.1.2. Syntax tree-based prosody generation module

The ANNs were always speaker-specific and trained on the above-mentioned adaptation data in one of the four target speakers' voices. By contrast, the mutator functions were estimated on various speech data:

- PhonDat1 and PhonDat2, with all the training speakers regarded as a single one (hereafter *system A*)
- the same adaptation data used to train the ANNs (hereafter *system B*)
- 1572 prosodically rich utterances in German read by a professional female speaker (hereafter *system C*)

The adaptation data contained merely 75 utterances per target speaker. It was indeed rather a small number. Apparently we should have trained the ANNs of systems A, B and C and the mutator functions of system B on much more speech data. The reasons why we did not do so were: 1) that system B sounded reasonably natural, which was beyond our expectation, and 2) that we could fairly compare the HMM-based speech synthesiser with our syntax tree-based module when they were "adapted" with the same amount of data from a target speaker.

4.2. Evaluations and Results

All the subjective evaluations were carried out in the form of listening tests (AB tests for naturalness assessment and ABX tests for speaker similarity assessment, with an option "equally natural/similar" given as well). Twelve listeners, who are native speakers of the German language, participated in every test. The speech samples (audio) presented to the listeners were generated as per a set of test sentences (text), and then some of the samples were randomly selected when a listener opened the web page for evaluation. Whiskers in the following figures indicate 95% confidence intervals given by t-test.

4.2.1. Naturalness: system A versus system B

Firstly, we compared the naturalness of system A with that of system B. Eight pairs of synthesised utterances in total (two pairs per target speaker) were played to a listener. The size of the test set was 33 sentences. The judgements from the 12 listeners are presented in Figure 3.

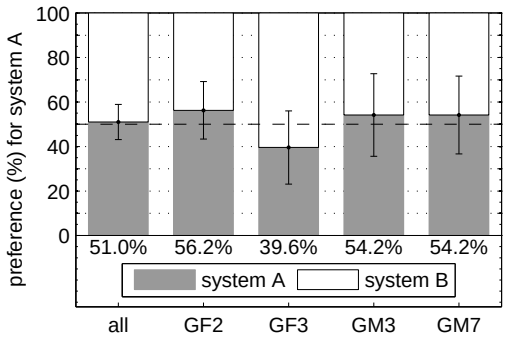

Figure 3: *naturalness assessment (A vs B)*

The only difference between system A and system B was the data set on which their mutator functions were trained. For system A, the data set was the large corpus combination of PhonDat1 and PhonDat2. This combination contained a lot of training speakers. More importantly, each prompt of PhonDat (1&2) was read by quite a few speakers, which is effectively a great advantage of training a speaker-independent model. For system B, the data set was the same as that on

which its speaker-specific ANNs were trained. Therefore system B was completely speaker-specific. Figure 3 conveys a message that system A and system B can be regarded as equally natural. Considering the difference between the two systems, we tend to believe that it is not necessary to train mutator functions and ANNs on speech data from the same target speaker. In other words, mutator functions should be considerably speaker-independent.

4.2.2. Naturalness: system A versus system C

Secondly, we compared the naturalness of system A with that of system C. Eight pairs of synthesised utterances in total (two pairs per target speaker) were played to a listener. The size of the test set was 33 sentences. The judgements from the 12 listeners are presented in Figure 4.

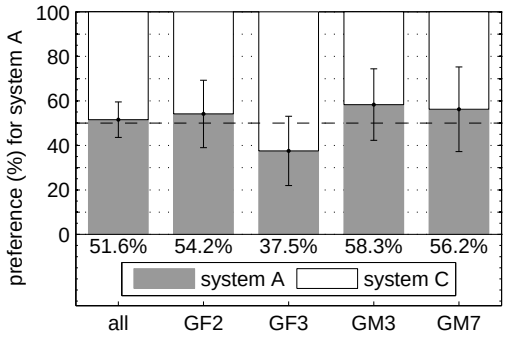

Figure 4: *naturalness assessment (A vs C)*

The only difference between system A and system C was also the data set on which their mutator functions were trained. For system C, the data set was collected from a totally different lady from any target speaker. However, system A and system C can be regarded as equally natural according to Figure 4. This strengthens our hypothesis that mutator functions mainly captures speaker-independent prosodic information because of their nature (i.e. mainly capturing high-level information).

4.2.3. Naturalness: system A versus system D

Thirdly, we compared the naturalness of system A with that of system D. Eight pairs of synthesised utterances in total (two pairs per target speaker) were played to a listener. The size of the test set was 33 sentences. The judgements from the 12 listeners are presented in Figure 5.

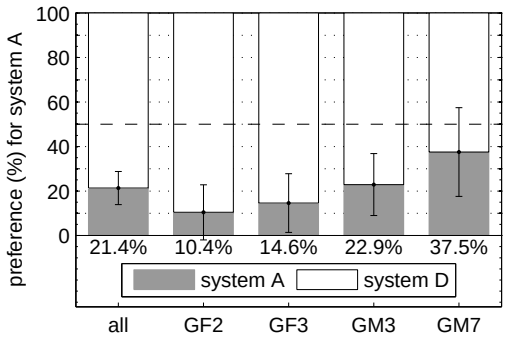

Figure 5: *naturalness assessment (A vs D)*

Figure 5 indicates that the conventional prosody prediction mechanism of HMM-based speech synthesis outperformed the syntax tree-based prosody generation module under our particular experimental conditions. This result is not very surprising. We have observed that in general the prosody prediction mechanism of HMM-based speech synthesis produces more fluctuating F_0. A possible reason is that such F_0 curves resulted in speech samples that were less robotic-sounding

to the listeners. Another possible reason is that the ANNs were trained on 75 adaptation utterances, while the HMMs were trained on PhonDat (1&2) and then adapted with the 75 utterances. This difference suggests that we should probably train ANNs on a large, multi-speaker corpus and then adapt them with a small amount of speech from a target speaker.

4.2.4. Speaker similarity: system A vs system E

The speaker similarity evaluation was a comparison between system A and an additional system, namely E, which was similar to system A but whose ANNs were trained on PhonDat (1&2) too. As a result, system A could be viewed as a special, "adapted" version of system E.

Eight pairs of synthesised utterances in total (two pairs per target speaker) were played to a listener. The size of the test set was 44 sentences from the EMIME bilingual corpus [17]. The reference sample for each pair was generated from natural prosodic features extracted from original recordings as well as synthetic non-prosodic features produced by the adapted HMM-based speech synthesiser. The judgements from the 12 listeners are presented in Figure 6.

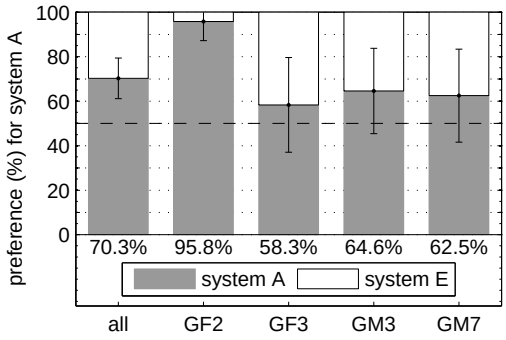

Figure 6: *speaker similarity assessment (A vs E)*

Spectral features were always generated by the adapted HMM-based speech synthesiser in our experiments. This means the listeners needed to focus on prosodic cues in the speaker similarity evaluation. Prosody is a "long-term" phenomenon. Adaptation data, which is normally a small set of utterances, is not likely to contain many local speaker-dependent prosodic cues. Hence Figure 6 should mostly reflect the effects of global speaker-dependent prosodic cues. It can be seen in Figure 6 that overall, global speaker-dependent prosodic cues were captured to different extents by the speaker-specific ANNs of system A, especially for target speaker GF2. We listened to the samples generated by system A and observed that GF2's pitch and speaking speed were both well reproduced by system A in her voice. Given that this EMIME bilingual corpus is in the neutral read style (thus fewer local speaker-dependent prosodic cues should be expected) and that the spectral features in a pair were generated by the same adapted HMM-based speech synthesiser, apparently global speaker-dependent prosodic cues were quite influential in speaker similarity.

5. Conclusions

Even though our syntax tree-based prosody generation module was developed on speech data in only one speaker's voice, we found its two-layer architecture made itself flexible in the sense that its mutator functions mainly captured speaker-independent prosodic information and could work well with its ANNs trained on speaker-specific speech data in a different voice. Apart from that, these ANNs were able to reproduce basic global speaker-dependent prosodic cues that were contained in a small amount of "adaptation data". The findings in this preliminary work would be of great help in building a *highly personalised* speech-to-speech translator in the future, even if the conventional prosody prediction mechanism of HMM-based speech synthesis outperformed the syntax tree-based prosody generation module in terms of the naturalness of synthesised prosody under our particular experimental conditions.

6. Acknowledgements

The research work presented in this technical report was funded by the SIWIS project (SNSF Grant CRSII2-141903). The authors would also like to thank all the anonymous native German speakers who participated in our listening test.

7. References

[1] Spoken Interaction with Interpretation in Switzerland (SIWIS). [Online]. Available: http://www.idiap.ch/project/siwis

[2] M. Wester, J. Dines, M. Gibson, H. Liang, Y.-J. Wu, L. Saheer, S. King, K. Oura, P. N. Garner, W. Byrne, Y. Guan, T. Hirsimäki, R. Karhila, M. Kurimo, M. Shannon, S. Shiota, J. Tian, K. Tokuda, and J. Yamagishi, "Speaker adaptation and the evaluation of speaker similarity in the EMIME speech-to-speech translation project," in *Proc. of 7th ISCA Workshop on Speech Synthesis*, Sep. 2010, pp. 192–197.

[3] M. Kurimo, W. Byrne, J. Dines, P. N. Garner, M. Gibson, Y. Guan, T. Hirsimäki, R. Karhila, S. King, H. Liang, K. Oura, L. Saheer, M. Shannon, S. Shiota, J. Tian, K. Tokuda, M. Wester, Y.-J. Wu, and J. Yamagishi, "Personalising speech-to-speech translation in the EMIME project," in *Proc. of the ACL 2010 System Demonstrations*, Jul. 2010, pp. 48–53.

[4] K. Tokuda, H. Zen, and A. W. Black, "An HMM-based speech synthesis system applied to English," in *Proc. of IEEE Workshop on Speech Synthesis*, Sep. 2002, pp. 227–230.

[5] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. of Eurospeech*, Sep. 1999, pp. 2347–2350.

[6] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, Nov. 2009.

[7] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation for HMM-based speech synthesis system using MLLR," in *Proc. of ESCA/COCOSDA Workshop on Speech Synthesis*, Nov. 1998, pp. 273–276.

[8] ——, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Proc. of ICASSP*, May 2001, pp. 805–808.

[9] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[10] H. Zen, N. Braunschweiler, S. Buchholz, M. J. F. Gales, K. Knill, S. Krstulović, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 6, pp. 1713–1724, Aug. 2012.

[11] C. Traber, "SVOX: The implementation of a text-to-speech system for German," Ph.D. dissertation, ETH Zürich, 1995.

[12] S. Hoffmann and B. Pfister, "Employing sentence structure: Syntax trees as prosody generators," in *Proc. of Interspeech*, Sep. 2012, pp. 470–473.

[13] T. Ewender, S. Hoffmann, and B. Pfister, "Nearly perfect detection of continuous F0 contour and frame classification for TTS synthesis," in *Proc. of Interspeech*, Sep. 2009, pp. 100–103.

[14] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman & Hall, 1984.

[15] C. Draxler, "Introduction to the Verbmobil-PhonDat database of spoken German," in *Proc. of the 3rd International Conference on the Practical Application of PROLOG*, Apr. 1995, pp. 201–212.

[16] J. Yamagishi and O. Watts, "The CSTR/EMIME HTS system for Blizzard Challenge 2010," in *Proc. of the Blizzard Challenge*, Sep. 2010.

[17] M. Wester, "The EMIME bilingual database," University of Edinburgh, Tech. Rep. EDI-INF-RR-1388, 2010.

[18] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, Apr. 1999.