TEMPLATE BASED TECHNIQUES FOR AUTOMATIC SEGMENTATION OF TTS UNIT DATABASE

S. Adithya[1†], *Sunil Rao*[2†], *C. Mahima*[4], *S. Vishnu*[3†], *Mythri Thippareddy*[4], *V. Ramasubramanian*[4*]

[1]University of California, San Diego
[2]Arizona State University, Tempe, Arizona
[3]San Diego State University, San Diego
[4]PES Institute of Technology - Bangalore South Campus (PESIT-BSC), Bangalore, India
[1]aseshasa@eng.ucsd.edu, [*]v.ramasubramanian@pes.edu

ABSTRACT

We address the problem of automatic segmentation of the unit database in unit-selection based TTS and propose template based forced alignment segmentation in the one-pass dynamic programming (DP) framework with several variants: i) multi-template representation derived by modified K-means (MKM) algorithm, ii) context-independent and context-dependent templates for reduced multi-template representation, iii) segmental K-means algorithm with MKM modeling of phone classes, as a template-based equivalent of the conventional embedded re-estimation procedure for HMM based modeling and segmentation, that is typical for deriving unit-databases for TTS (e.g. EHMM in Festival). We first benchmark the performance of the proposed segmentation framework on TIMIT database for phonetic segmentation given the availability of phonetic labeling ground truth in TIMIT. We then apply the proposed template based segmentation algorithms for syllabic Indian language TTS, and benchmark the proposed segmentation using objective measures based on spectral distortions (SD) obtained on time-aligned speech utterances and compare it with other recent segmentation approaches, namely the group-delay (GD) based semi-automatic method, Hybrid method, EHMM, HMM and SKM-HMM and show that the proposed template based approaches offer comparable and better spectral distortions, validating their ability to provide accurate high-resolution segmentation of the unit-database.

Index Terms— Unit-database segmentation, TTS, Template-based segmentation, one-pass DP, segmental K-means

1. INTRODUCTION

Segmentation of continuous speech in a large speech corpus is of primary importance in unit-selection based concatenative text-to-speech (TTS) systems. Since this framework of TTS relies on units (e.g. phones, diphones, syllables) that are concatenated for synthesis, the accuracy with which the unit database (typically, a single-speaker, large speech corpus of 5 to 10 hours) is segmented and labeled determines the quality of synthesis. A poor segmentation, wherein the unit boundaries are incorrect with respect to the actual (underlying ground truth) boundaries even by small margins, can lead to poor quality of the synthesized speech, reflecting in terms of poorly articulated units, spurious phonetic intrusions (a unit containing parts of neighboring phonetic class) and unnatural durations. In this paper, we proposed high resolution template based segmentation techniques, and compare these with other conventional segmentation methods for TTS unit-database segmentation, including HMM based techniques, bringing out the salient differences.

2. RELATION TO PRIOR WORK

The problem of accurate segmentation of the unit database has attracted good attention, with various solutions to date, e.g. [1], [2]. The most typical of these is the embedded re-estimation based segmentation, which employs the HMM parameter estimation procedure [3] from continuous speech with only the word level transcription of the corpus, combined with a pronunciation dictionary or a grapheme-to-phoneme converter (G2P). The iterative procedure estimates the HMM parameters of the underlying units of speech reliably even while yielding the segmentation of the speech corpus in terms of these units. This method is termed EHMM in the Festival TTS platform [4].

More recently, as part of the TTS Consortium for Indian Languages [5], [6] a hybrid segmentation (Hybrid) procedure was proposed [7], which performs an iterative refinement of an initial group-delay based syllabic segmentation of the speech corpus. This employs a syllable conditioned embedded re-estimation of the HMMs of the units, and arrives at both phonetic (called mono-phone) and syllabic boundaries. This segmentation has since been used to derive the unit-database for 8 Indian languages and also used for 6 Indian languages in the Blizzard Challenge 2014 [8].

In this paper, we address the problem of automatic segmentation of the unit database into phonetic and syllabic units using a template based approach, motivated by various aspects: i) the existing EHMM based approach [4] as well as the hybrid segmentation (Hybrid) approach [7] are based on HMM based modeling of the units. While these are quite adequate, they need substantial instances of each unit for the HMMs to be trained reliably, and in turn yield a good segmentation. Secondly, it is not clear whether HMMs, being statistical models can provide high-resolution segmentation due to its inherent probabilistic nature, which models a speech unit by a small number of state conditioned pdfs. ii) An alternative to HMM is to use templates as representation of the units, a paradigm which has received some attention in the recent years - traditionally in low bit-rate speech coding [10, 11, 12, 13, 14, 15, 16, 17], in speech recognition [18, 19], speaker-recognition [9], audio-analytics [20] and human action indexing from video [21, 22].

Templates are a non-parametric model of a unit class and for the purpose of segmentation, represent the temporal content of a speech unit in a high-resolution, over-sampled manner (the sequence of feature vectors retained as it is) and can yield high degree of matching to new test data, and hence have a potential to yield high-

[†] Authors carried out this work at PESIT-BSC, Bangalore as Research Associates in the TTS-Consortium project.

resolution segmentation. Template based modeling can also be expected to require far less instances per unit for a given segmentation performance, in comparison to larger number of instances per unit for HMM training. Despite the above cited work which have explored the use of templates for speech coding, speech recognition and speaker recognition, templates have not been examined for the segmentation problem in speech synthesis. Specifically, while some of the above cited work emphasized the use of templates in the form of deriving segment codebooks or template modeling [10, 11, 12, 14, 15, 16, 17, 18, 19, 20], none of them addressed the issue of template based approaches on the accuracy of an incidental segmentation of the underlying speech corpus, something which is critical in a synthesis context. A more general overview of several early classes of segmentation techniques can be found in [23].

In this paper, we propose a one-pass dynamic programming (DP) based forced alignment using template models of a phonetic class. While one-pass DP algorithm is well known as a connected word recognition algorithm [24], we use it in forced alignment mode here with several variants. As a first variant ('Template' method), we use a small seed data comprising a small number of templates per phone class (e.g. 10) to segment a large corpus, and examine the segmentation error in a controlled manner with respect to ground truth segmentation of the corpus. Here, we examine various issues of template modeling such as the number of templates needed per phone class, how to extract them by a modified K-means (MKM) algorithm, use of context-independent and context-dependent templates and their impact on the number of templates needed per class. Within this algorithmic framework, we also propose a segmental K-means (SKM) algorithm ('SKM-Template') as a template equivalent of the conventional embedded re-estimation procedure for HMM training and segmentation. This algorithm is an iterative realization of two steps - a forced alignment by multi-template one-pass DP segmentation and a modified K-means algorithm to derive a small set of template codebook per phone class from the forced alignment segmentation clusters - starting with flat start, and hence representing a 'seed-less' procedure for template based segmentation.

We benchmark all of the above template based methods using the TIMIT database, with the availability of ground truth segmentation. We then apply the proposed template based techniques ('Template' and 'SKM-Template') to segmentation of Indian languages along with 5 other segmentation algorithms, namely, the group-delay based semi-automatic algorithm (GD) [7], hybrid segmentation (Hybrid) [7] as indicated above and Festival's EHMM [4], HMM, SKM-HMM. We show relative performance measures of these 7 segmentation techniques (Template, SKM-Template, GD, Hybrid, EHMM, HMM, SKM-HMM) in terms of segmentation error statistics and double-ended spectral distortion based objective TTS quality measure [26]. Through these measures, we show that the proposed template based approaches have comparable or even better quality than the other approaches, thereby validating their advantages such as limited modeling data given the small number of seed templates needed to perform a segmentation of a large corpus, high-resolution segmentation afforded by the inherent frame-level representation of a unit, and the remarkably accurate segmentation of the more general seed-less flat-start template based 'SKM-Template' procedure.

3. PROPOSED TEMPLATE BASED SEGMENTATION

3.1. Forced alignment using one-pass DP

The problem of segmentation is best addressed and solved in a forced alignment framework, where the transcription of the speech in terms of the desired units is assumed or made available. In this work, we first benchmark the performance of the proposed segmentation algorithm using TIMIT, and hence have used phonetic classes of the TIMIT database as the units of segmentation.

Consider an input speech utterance to be segmented, in the form of a sequence of feature vectors, say MFCCs, $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T$, and its unit-level transcription yields a sequence of V units $1, \ldots, v, \ldots, V$. For each unit v, we use a number of templates (say, M) to model the unit class, i.e., $R_{vm}, m = 1, \ldots, M$ for unit v. An one-pass dynamic programming (DP) algorithm, normally used for connected word recognition with template models of a word, can now be adapted to perform forced alignment using M templates/unit, i.e., $\{R_{vm}, m = 1, \ldots, M, v = 1, \ldots, V$ on the y-axis, against the input feature vector sequence $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T$ on the x-axis. Such a realization was actually proposed earlier [9], in the context of text-dependent speaker recognition, but with the forced alignment being done on sequence of words comprising a password text of short sequences of words. Here, the algorithmic realization is essentially the same, except with the primary difference being the use of much longer sequences of shorter phonetic units.

The one-pass DP forced alignment has two recursions - the within-unit recursion and across-unit recursion, as given below. The within-unit recursion is applied to all frames of all templates of each unit, except the first frame of all templates of each unit. It builds a path for all such frames within a unit being reached from the interior frames at the immediately past input frame. The across-unit recursion is applied to the first frame of every template of every unit (from the second unit onwards), so as to receive a across unit transition from the last frame of any template of the preceding unit in the unit transcription of the input utterance.

The general equations for these two types of recursions are:

Within-unit recursion

$$\begin{aligned} D(t,n,v) &= d(t,n,v) + \min\{D(t-1,n,v), \\ &\qquad D(t-1,n-1,v), D(t,n-1,v)\} \end{aligned} \tag{1}$$

Across-unit recursion

$$\begin{aligned} D(t,1,v) &= d(t,1,v) + \min\{D(t-1,1,v), \\ &\qquad \min_{u \in Pred(v)} D(t-1,N_u,u)\} \end{aligned} \tag{2}$$

Here, $D(t,n,v)$ is the minimum accumulated distortion by any path reaching the grid point defined as frame 'n' of unit-template 'v' and frame 't' of the input utterance; $d(t,n,v)$ is the local distance between the n^{th} frame of unit-v template and t^{th} frame of the input utterance. The within-unit recursion applies to all frames of a unit v template, which are not the starting frame (i.e., $n > 1$). The across-unit recursion applies to frame 1 of any unit-v to account for a potential 'entry' into unit v template from the last frame N_u of any of the templates u of the unit $v-1$ which is a predecessor of unit-v; i.e., denoted as $u \in Pred(v)$; these are the valid predecessors of any unit v consisting of the multiple templates $Pred(v)$ of the unit $v-1$ preceding the unit v in the unit-level transcription of the input utterance; for instance, in an utterance with transcript /s//u//t/ and $v = /u/$, then $Pred(v = /u/) = R_{s1}, R_{s2}, \ldots, R_{sM}$; likewise, $Pred(v = /t/) = R_{u1}, R_{u2}, \ldots, R_{uM}$. This across-unit recursion takes care of entry into any template of any unit from any template of any preceding unit in the unit transcription.

The above recursions are applied to all frames of all units in the unit-level transcript of the input utterance for every frame $t = 1, \ldots, T$ of the input utterance $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T$. The forced alignment stops at $t = T$ and backtracks from the trellis co-ordinate (T, N_{m^*}, m^*), where $m^* = \arg\min_{m=1,\ldots,M} D(T, N_{V_m}, V_m)$, V_m being the m^{th} template of the last unit V with N_{V_m} frames.

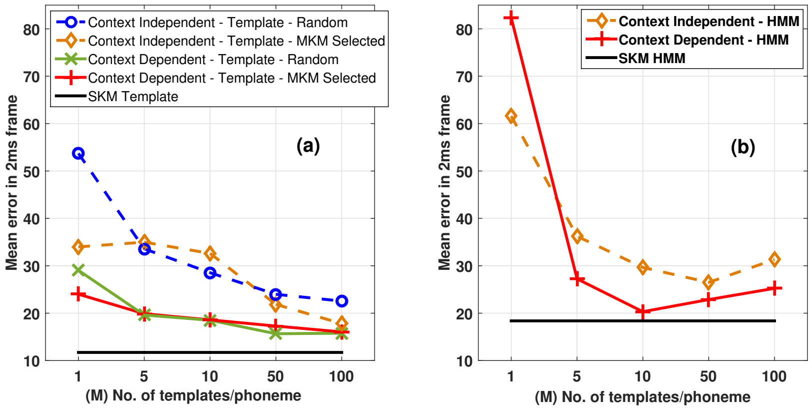

Fig. 1. *Mean error of segmentation vs number templates/unit (M) for (a) Template-based and (b) HMM-based segmentation methods*

The best path retrieved by such a backtracking yields the desired segmentation of the input utterance $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, \ldots, \mathbf{o}_T$ into V units with corresponding segment boundaries, which are then compared with the ground-truth boundaries for obtaining the statistics of the segmentation error or used as the segmented unit-database for further unit-selection synthesis.

4. TEMPLATE VS HMM BASED SEGMENTATION

Here, we present the performance of the proposed template based segmentation in Fig. 1(a) and compare it with HMM based segmentation in Fig. 1(b) in terms of mean error of segmentation (measured in number of 2ms frames) with respect to TIMIT ground truth (on the y-axis) vs the number of templates/phoneme used in the forced alignment (on the x-axis), i.e., M as defined above in Sec. 3.1 for $M = 1, 5, 10, 50, 100$. These results are obtained for segmenting 1000 TIMIT sentences (125 speakers with 8 sentences per speakers excluding the common sa1 and sa2 sentences to avoid biases). For the template based segmentation in Fig. 1(a), we consider different cases of phoneme template definition: i) context-independent and ii) context-dependent templates, selected randomly or by the modified K-means algorithm [25]. In the case of HMM based segmentation in Fig. 1(b), the number of templates/phoneme (on the x-axis) M corresponds to the number of templates used for training the HMM (either context-independent or context-dependent). The templates used for such a training are identical to the templates used in the template-based segmentation in Fig. 1(a). HMMs used are 3 state left-to-right models, with 1-5 mixture/state, depending on M, used in forced alignment mode Viterbi for segmentation. The results corresponding to 'SKM-Template' and 'SKM-HMM' in these figures are discussed in Sec. 5.

The following can be noted: i) in Fig. 1(a), the context dependent templates offer significantly lower mean errors (down to 30ms) for much smaller number of templates / phoneme-unit, ii) MKM selection is seen to be conducive to the extent of eliminating the variability that is inherent in a random selection. Considering Fig. 1(b), the following can be noted: i) context-dependent HMMs perform better than context-independent HMMs, keeping with the reduced variability in the modeled set of templates/phoneme, ii) HMM based segmentation can be noted to have higher mean-error for small number of templates/phoneme ($M = 1, 5$) than the template-based segmentation. Note that the HMMs in Fig. 1(b) correspond to the MKM cases of Fig. 1(a), as MKM selected templates represent non-parametric modeling of a phoneme class, iii) As M increases to 10, HMMs generalize better and the mean-error reduces at par with template-based segmentation. iv) the increasing mean-error for larger number of templates for HMM can be attributed to the HMMs requiring more mixtures/state to adequately model the increased variability present in the training data per phoneme class and further generalize well on unseen data when used for segmentation. With these results, we are able to conclude that template-based segmentation offers better segmentation performance than HMMs, when the number of templates/phoneme are very small, with progressively improving performance for larger number of templates/phoneme, and that HMMs offer comparable performance only when larger number of training templates are available.

5. EHMM AND SEGMENTAL K-MEANS (SKM)

In Sec. 2 we referred to the conventional embedded re-estimation procedure for HMM training and segmentation (EHMM). As an alternative to this, we propose here a segmental K-means algorithm for both template based representation and HMM based representation, i.e., SKM-Template and SKM-HMM respectively, which are iterative realizations of two steps: i) a forced alignment by multi-template one-pass DP segmentation or HMM based Viterbi segmentation, and ii) a modified K-means algorithm [25] to derive a small set of template codebook per phone class from the forced alignment segmentation clusters in SKM-Template or HMMs trained from the forced alignment clusters in SKM-HMM. Like the EHMM, the SKM starts with flat start, and represents a 'seed-less' procedure for template based segmentation (unlike the method in Sec. 3.1 which uses a small seed). By this, we realize an accurate modeling of the underlying phonetic units, and further yield accurate segmentation due to the iterative nature of refining the templates/HMM modeling the units and the successive forced alignments.

The convergence characteristic of the SKM-Template algorithm is shown in Fig. 2(a) and the corresponding consistent reduction in the mean segmentation error (and associated lowering σ bar) in Fig. 2(b) on 1000 TIMIT sentences. At convergence, the SKM-Template algorithm yields a remarkably low mean error of 1.2 20ms frames (24 ms) and an associated low σ. Interestingly, this 'seedless' SKM-Template with mean-error of 24 ms, outperforms the 'seeded' cases shown in Fig. 1(a), thereby validating the effectiveness of the iterative procedure to improve the segmentation even starting from flat-start (marked as iteration 0 in Fig. 2(b). More importantly, note also that the SKM-Template (black-solid line) in Fig. 1(a) outperforms the SKM-HMM (black-solid line) in Fig. 1(b) by a good margin.

6. SYLLABIC SEGMENTATION

The above template based methods (referred to as 'Template' and 'SKM-Template') were benchmarked using the TIMIT database,

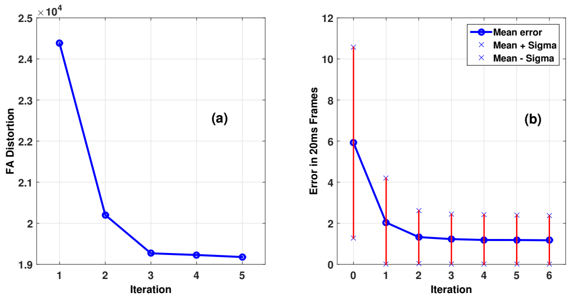

Fig. 2. *SKM-Template performance: (a) FA distortion vs iteration; (b) Mean error in 20 ms frames vs iteration with σ bars*

owing to the availability of ground truth segmentation. We now apply the same techniques to a 'syllabic' segmentation of one Indian language (namely, Tamil (TA), among the 13 languages of the TTS Consortium [5], [6]) along with 5 other segmentation algorithms, namely, the group-delay based semi-automatic algorithm (GD) [7], hybrid segmentation (Hybrid) [7], Festival's EHMM [4], HMM based segmentation (HMM10, with $M = 10$) in Sec. 4 and segmental K-means HMM (SKM-HMM) in Sec. 5. Since the GD method is a semi-automatic procedure with relatively high accuracy (involving manual correction of the syllabic units determined by a first-pass group-delay based automatic segmentation), we use GD as ground-truth and measure the segmentation mean-error of the other 6 methods (EHMM, Hybrid, Template, SKM-Template, HMM10 and SKM-HMM methods proposed here) with respect to this ground-truth. Fig. 3 shows the mean-error for these 6 methods; it can be noted that while 'Hybrid' performs best, the 'Template' and 'SKM-Template' based methods offer the next best performance, while EHMM performs poorer, and HMM10 and SKM-HMM are the worst. The 'Template' method uses a very small number of seeds (10 templates/monophone unit) of TIMIT phone-like units followed by a syllabic grouping of these units using a reverse syllable dictionary. Considering this small seeding (when compared to a more extensive syllable conditioned HMM re-estimation procedure of 'Hybrid'), it can be noted that the 'Template' method does offer an effective performance along with the 'seed-less' and more attractive 'SKM-Template' method.

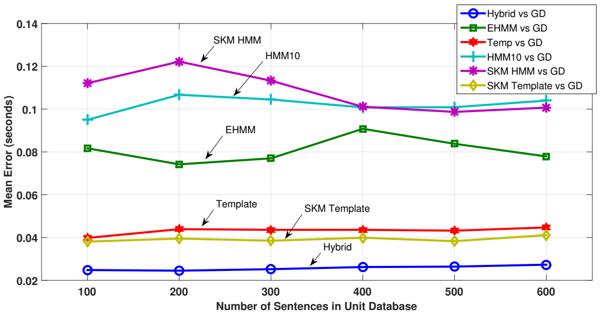

Fig. 3. *Mean syllabic segmentation error of different segmentation techniques*

7. SYNTHESIS AND DOUBLE-ENDED MEASURE

Here we characterize the performance of the different segmentation techniques in terms of actual TTS performance. In order to quantify the differences in the synthesized speech for the different segmentation techniques, without being effected by subjective measures such as MOS, DMOS, WER etc., we compare the 7 different segmentation techniques (GD, Hybrid, Template, SKM-Template, EHMM, HMM10, SKM-HMM) by employing a 'double-ended' quality measure we had proposed recently [26]. Here, we compute the spectral distortion (SD) [27, 28] between speech synthesized from text which has a corresponding reference speech by the same speaker as the unit database. In order to account for non-linear temporal variability between these two speech (one spoken by the speaker, and the other with durations of units as determined from text and as occurring in the unit-database without prosody modification), the two speech are time-aligned by dynamic time-warping; this accounts for their intrinsic durational and speaking rate differences. Fig. 4 shows the mean SD for all the above 7 methods using a unit-database size of 600 sentences (corresponding to 600 in Fig. 3), for 50 in-database sentences (within the unit-database, marked as IN-DB in the x-axis) as calibration and for 50 out-of-database sentences (marked as OUT-DB in the x-axis) - which reflects the actual performance of arbitrary text as input to the TTS system. It can be seen that the 'IN-DB' SDs are at $\sim$1.5dB, (with 1dB corresponding to the 'transparent quality' quantization as known in speech coding [27, 28]), as is expected for in-database sentences, and that the 'OUT-DB' SDs are much higher ($\sim$4dB), though all 7 methods seem to 'bunch' together, indicating comparable quality (with GD and Hybrid showing the best baseline performance for IN-DB). We see that the proposed approaches 'Template' and 'SKM-Template' are effective in their performance, having comparable or even better quality than the other approaches (e.g. HMM10), thereby validating their advantages such as limited modeling data given the small number of seed templates needed to perform a segmentation of a large corpus, high-resolution segmentation afforded by the inherent frame-level representation of a unit, as well as the distinctly superior performance of the more general 'seed-less' flat-start 'SKM-Template' procedure.

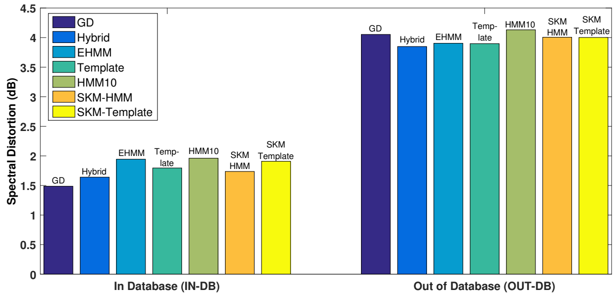

Fig. 4. *Spectral distortion for in-database (IN-DB) and out-of-database (OUT-DB) sentences for the 7 segmentation techniques*

8. CONCLUSIONS

We have proposed template based techniques for automatic segmentation of TTS unit databases. We have proposed 1-pass DP based forced alignment segmentation methods using multi-templates, derived as context-independent or context-dependent ones, via a modified K-means algorithm, as well as a segmental K-means algorithm for a seed-less segmentation. We have benchmarked the performance of these algorithms on TIMIT database, with phonetic ground truth segmentation, and also applied it for syllabic segmentation of an Indian language database, and compared their performance with 5 other segmentation techniques, in terms of spectral distortion as an objective measure, and shown that the proposed methods are as effective or better than the other more complex algorithms.

9. ACKNOWLEDGMENTS

The authors thank the Department of Information Technology, Ministry of Communication and Technology, Government of India, for funding the TTS Consortium project, "Development of Text-to-Speech synthesis for Indian Languages Phase II", Ref. no. 11(7)/2011HCC(TDIL).

10. REFERENCES

[1] A. W. Black and J. Kominek. Optimizing segment label boundaries for statistical speech synthesis. Proc. ICASSP '09, pp. 3785–3788, 2009.

[2] A. Sethy and S. S. Narayanan. Refined speech segmentation for concatenative speech synthesis. Proc. Interspeech '02, pp. 149-152, 2002.

[3] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. The HTK Book (for HTK Version 3.4). Cambridge University Engineering Department, 2002.

[4] A. Black, P. Taylor, and R. Caley, The Festival speech synthesis system. http://festvox.org/festival/, 1998.

[5] B. Ramani, S. Lilly Christina, G. Anushiya Rachel, V. Sherlin Solomi, M. K. Nandwana, A. Prakash, S. Aswin Shanmugam, R. Krishnan, S. Kishore, K. Samudravijaya, P. Vijayalakshmi, T. Nagarajan, and H. A. Murthy. A common attribute based unified HTS framework for speech synthesis in Indian languages. Proc. SSW8, 2013, pp. 291-296.

[6] H. Patil, T. Patel, N. Shah, H. Sailor, R. Krishnan, G. Kasthuri, T. Nagarajan, L. Christina, N. Kumar, V. Raghavendra, S. Kishore, S. Prasanna, N. Adiga, S. Singh, K. Anand, P. Kumar, B. Singh, S. Binil Kumar, T. Bhadran, T. Sajini, A. Saha, T. Basu, K. Rao, N. Narendra, A. Sao, R. Kumar, P. Talukdar, P. Acharyaa, S. Chandra, S. Lata, and H. Murthy. A syllable-based framework for unit selection synthesis in 13 Indian languages. Proc. Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), 2013 International Conference, pp. 1-8, Nov. 2013.

[7] S. Aswin Shanmugam, Hema Murthy. A hybrid approach to segmentation of speech using group delay processing and HMM based embedded reestimation. Proc. Interspeech '14, pp. 1648–1652, Singapore, Sep 2014.

[8] K. Raghava Krishnan, S. Aswin Shanmugam, Anusha Prakash, Kasthuri G R and Hema A Murthy. IIT Madras's Submission to the Blizzard Challenge 2014. Proc. Blizzard Challenge 2014, Satellite workshop of Interspeech '14, Singapore, 2014 (http://festvox.org/blizzard/blizzard2014.html)

[9] V. Ramasubramanian, Amitav Das, and V. Praveen Kumar. Text-dependent speaker-recognition using one-pass dynamic programming algorithm. In Proc. ICASSP06, pp. I-901-I-904, Toulouse, France, May 2006.

[10] S. Roucos, R. Schwartz, J. Makhoul. Segment quantization for very-low-rate speech coding. Proc. ICASSP '82, vol. 3, pp. 1565–1568, Paris, France, 1982.

[11] S. Roucous, R.M. Schwartz, J. Makhoul. A segment vocoder at 150 b/s. Proc. ICASSP 83, pp. 61-64, Boston, 1983.

[12] Y. Shiraki, M. Honda. LPC speech coding based on variable-length segment quantization. IEEE Trans. Acoust. Speech Signal Process. 36(9), pp. 1437-1444, 1988.

[13] T. Svendsen, F.K. Soong. On the automatic segmentation of speech signals. Proc. ICASSP 87, pp. 77-80, 1987.

[14] T. Svendsen. Segmental quantization of speech spectral information. Proc. ICASSP 94, 1994, pp. I-517-I-520, 1994.

[15] V. Ramasubramanian, T.V. Sreenivas Automatically derived units for segment vocoders. Proc. ICASSP 04, pp. 473–476, Montreal, Canada, 2004.

[16] V. Ramasubramanian. Ultra low bit-rate speech coding: An overview and recent results Proc. IEEE SPCOM, Indian Institute of Science, Bangalore, 2012

[17] V. Ramasubramanian and Harish Doddala. Ultra low bit-rate speech coding. SpringerBriefs in Speech Technology, Springer, 2015.

[18] M. DeWachter, M.Matto, K. Demuynck, P.Wambacq, R. Cools and D. Van Compernolle. Template-based continuous speech recognition. In IEEE Transactions on Audio, Speech and Language Processing, pp. 1377–1390, vol. 15, no. 4, May 2007.

[19] V. Ramasubramanian, Kaustubh Kulkarni and Bernhard Kaemmerer. Acoustic modeling by phoneme templates and one-pass DP decoding for continuous speech recognition. In Proc. ICASSP '08, pp. 4105–4108, Las Vegas, Mar 2008.

[20] Srikanth Cherla and V. Ramasubramanian. Audio analytics by template modeling and 1-pass DP based decoding. Proc. Interspeech-2010, pp. 2230–2233, Chiba, Japan, Sep 2010.

[21] Srikanth Cherla, Kaustubh Kulkarni, Amit Kale, V. Ramasubramanian. Towards Fast, View Invariant Human Action Recognition. Proc. IEEE Workshop for Human Communicative Behavior Analysis at CVPR 2008, Anchorage, Alaska, Aug 2008.

[22] Kaustubh Kulkarni, Srikanth Cherla, Amit Kale, V. Ramasubramanian. A Framework for Indexing Human Actions in Video. Proc. 1st International Workshop on Machine Learning for Vision-Based Motion Analysis at ECCV 2008, Marseille, France, Oct 2008.

[23] E. Vidal, A. Marzal. A review and new approaches for automatic segmentation of speech signals. in Signal Processing V: Theories and Applications, ed. by L. Torres, E. Masgrau, M.A. Lagunas (Elsevier Science Publisher B.V, Amsterdam), pp. 43-53, 1990.

[24] H. Ney. The use of one-stage dynamic programming algorithm for connected word recognition. IEEE Trans. on Acoust., Speech and Signal Proc, 32(2):263-271, Apr 1984

[25] J. G. Wilpon and L. R. Rabiner. A modified K-means clustering algorithm for use in isolated word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-23, No. 3, June 1985.

[26] Sunil Rao, C. Mahima, S. Vishnu, S. Adithya, A. Sricharan and V. Ramasubramanian. TTS evaluation: Double-ended objective quality measures. Proc. IEEE CONECCT 15, Bangalore, 2015

[27] K.K. Paliwal, B.S. Atal. Efficient vector quantization of LPC parameters at 24 bits/ frame. IEEE Trans. Speech Audio Process. 1, 3-14, 1993.

[28] K.K. Paliwal, W.B. Kleijn, Quantization of LPC parameters. In Speech Coding and Synthesis, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam), pp. 433-466. Chapter 12, 1995.