A FAST TABLE LOOKUP BASED, STATISTICAL MODEL DRIVEN NON-UNIFORM UNIT SELECTION TTS

*Yao Qian Frank. K. Soong *Xiaobo Zhou *Yundi Qian *Xiaotian Zhang

Microsoft Research Asia, Beijing, China
{yaoqian, frankkps}@microsoft.com

ABSTRACT

For multi-channel TTS applications, e.g. in a cloud service, it is highly desirable that high quality speech can be synthesized in low complexity. In this paper, we propose a fast table lookup based, statistical model driven approach to non-uniform unit selection TTS for that purpose. In TTS training, the voice font of all waveform segments is organized as a Gaussian kernel coded hash table and a table for storing quantized costs of all possible concatenation segment pairs. In synthesis, waveform segments with non-uniform lengths are first selected to construct a candidate lattice by looking up the Gaussian kernel coded hash table, and the best path is searched in the lattice by minimizing the accumulated concatenation scores, which are retrieved from the quantization table for possible concatenations. Experimental results show that the new approach can significantly reduce the search complexity while keep a high TTS voice quality.

Index Terms— statistical parametric synthesis, unit-selection based TTS, Hybrid TTS, voice font quantization, fast TTS

1. INTRODUCTION

In past few years, hybrid approaches to TTS [1-7] by combining parametric HMM and waveform unit selection have shown they can yield high synthesized voice quality in both naturalness and intelligibility. HMM, a parametric source-filter based model can yield smooth and highly intelligible TTS speech but still perceived as a traditional vocoder with a slight machine flavor [8]. On the other hand, the waveform concatenation-based unit selection TTS can yield fairly natural sounding speech but produce occasional undesirable concatenation glitches. The hybrid approaches, which use HMM to guide the unit selection process to minimize the spectral, pitch and duration mismatch and concatenation distortions, tend to preserve the advantages of both approaches [5]. A probabilistic criterion of likelihood [1], Kullback-Leibler divergence (KLD) between target and candidate phone-based HMMs [2] and the generated parameter trajectories from HMMs [3,4] are used to select the potential waveform unit candidates. An in-depth review has been given by Zen et al [5]. The unit selection oriented approach can also improve the quality of HMM-based synthesis by employing stable regions of natural units [6] or using the optimal rich context model sequences [7] to alleviate the sound muffling effects caused by overly smoothed HMM parameters due to the "averaging" process in both HMM training and synthesis [9] by maximizing the likelihood.

Recently, we proposed a trajectory tiling based approach to high quality speech synthesis [4], which uses the trajectories generated by the HMM to guide unit selection for finding the best match in spectrum, pitch and duration. The approach can render natural sounding speech without sacrificing the intrinsic high intelligibility of the HMM-based TTS, and has been confirmed in the Blizzard Challenge 2010 [4,21]. However, the computational complexity of the proposed approach is still very high, due to the parameter generation, distance calculation between guided trajectories and candidate trajectories, and maximization of the normalized cross-correlations (NCCs) for searching the best path in lattice and the optimal concatenation time instants [10]. In sever-based applications like cloud services, a TTS engine needs to manage multi-thread, multi-voice for multi-access with low complexity but without voice quality degradation. In this paper, we propose a fast, statistical model driven approach to non-uniform unit selection based TTS. It can achieve very fast synthesis by simple table lookup without degrading the synthesized voice quality.

2. OUR NEW APPROACH

The block diagram of our new TTS is shown in Fig. 1. In training, the speech signal is converted to a sequence of observed feature vectors and then modeled as a sequence of HMM states. Each state of HMMs is parameterized as a Gaussian distribution over the possible output. The entire speech database is coded by the mapped Gaussian kernels. In synthesis, input text is converted first into a sequence of contextual labels by the text analysis. The corresponding contextual state sequence is then generated. The waveform segment candidates are obtained from Gaussian kernel coded database. With the concatenation score table, the best candidate path is searched by the Viterbi algorithm to generate the final output speech.

2.1. Statistical Training of HMMs

Spectral envelope, fundamental frequency, and duration are modeled simultaneously by the corresponding HMMs first in maximum likelihood (ML) sense [11] and then refined to reduce synthesis errors of training sentence trajectories in the minimum generation error (MGE) sense [12]. Context-dependent phone models are used to capture the phonetic and prosody co-articulation phenomena. State typing based on decision-tree is used to alleviate insufficient training data problem. After HMMs training, the whole training data are firstly force-aligned by ML

*Work performed as an intern in the Speech Group, Microsoft Research Asia

criterion at the state level and then each state boundary is refined by the MGE criterion.

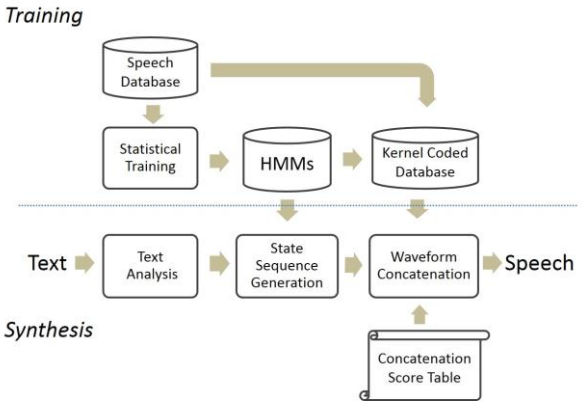

Fig. 1 Block diagram of the new table lookup, statistical model driven, non-uniform unit selection TTS.

2.2. Gaussian Kernel Coded Speech Database

In HMM training, stream-dependent models are built to cluster the spectral and pitch features into separated decision trees. The leaf nodes of decision three are used to quantize the force-aligned speech database. Each state-length waveform segment is coded by a tied spectral state ID and a tied F0 state ID. We also build up tied-sated distance tables for all spectral and pitch states. Kullback-Leibler Divergence (KLD) [18] is used to measure the similarity between two states of HMMs [19]. KLD is an information-theoretic measure of (dis)similarity between two probability distributions. For two given distributions, P and Q, of continuous random variables, the symmetric form of KLD between P and Q is:

$$D_{KL}(P,Q) = \int p(x)\log\frac{p(x)}{q(x)}dx + \int q(x)\log\frac{q(x)}{p(x)}dx \quad (1)$$

where p and q denote the probability density function (pdf) of P and Q. The spectral state has a multivariate single Gaussian distribution. The KLD between two spectral states has a closed form:

$$D_{KL}(P,Q) = \frac{1}{2}tr\{(\boldsymbol{\Sigma}_p^{-1}+\boldsymbol{\Sigma}_q^{-1})(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^T + \boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_q^{-1}+\boldsymbol{\Sigma}_q\boldsymbol{\Sigma}_p^{-1}-2\mathbf{I}\} \quad (2)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the corresponding mean vector and covariance matrix, respectively. Pitch features are modeled by an MSD-HMM, where two, discrete and continuous, probability spaces are modeled for unvoiced regions and voiced F0 contours [13], respectively. The upper bound of KLD between two states of MSD-HMMs can be derived as [14]:

$$\begin{aligned} D_{KL}(P,Q) \le{}& (w_0^p - w_0^q)\log\frac{w_0^p}{w_0^q} + (w_1^p - w_1^q)\log\frac{w_1^p}{w_1^q} \\ &+ \frac{1}{2}tr\{(w_1^p\boldsymbol{\Sigma}_p^{-1}+w_1^q\boldsymbol{\Sigma}_q^{-1})(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^T \\ &+ w_1^p(\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_q^{-1}-\mathbf{I}) + w_1^q(\boldsymbol{\Sigma}_q\boldsymbol{\Sigma}_p^{-1}-\mathbf{I})\} \\ &+ \frac{1}{2}(w_1^q - w_1^p)\log\left|\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_q^{-1}\right| \end{aligned} \quad (3)$$

where w_0 and w_1 are the prior probabilities of unvoiced and voiced subspaces, respectively.

With a Gaussian kernel coded database, we can construct a hash table to reduce search complexity via a quick table lookup. The key of hash table is the tied state ID and the corresponding value is the waveform segment index. Minimum description length (MDL) criterion [15] for balancing model complexity and training data size is used as a stop criterion for state clustering in growing the decision tree. The number of waveform segments clustered in each leaf node of the decision tree varies. In order to have adequate number of segment candidates in searching a best path in candidate lattice for the final concatenation, when the number of values for each hash key is less than a preset threshold, we group wave segments of other leaf node(s), which are similar, into the same hash key. Accordingly, the value of each hash key is associated with a distance, which indicates the dis-similarity of waveform segment to tied-state ID (key). A schematic illustration of hash table and waveform segment index table is shown in Fig. 2.

key	value1	value2	value3	value4
a_lsp_s2_1	50, 0	190, 0	191, 0.1	...
a_lsp_s2_2	191,0	695,0	...	...
		...		

ID	SentNo	StartFrame	Duration	LspState	F0State
0	1	0	18	SIL_lsp_s2_58	logF0_s2_434
1	1	18	1	SIL_lsp_s3_28	logF0_s3_230
...	...	...	...	...	...
50	1	100	2	a_lsp_s2_1	logF0_s2_434
...	...	...	...	...	...
190	2	230	12	a_lsp_s2_1	logF0_s2_434
191	3	12	3	a_lsp_s2_2	logF0_s2_856
...	...	...	...	...	...
695	4	107	4	a_lsp_s2_2	logF0_s2_856
...	...	...	...	...	...

Fig. 2 A schematic illustration of hash table and waveform segment index table.

2.3. Non-uniform Segment Selection

In synthesis phase, tied state sequence is generated by traversing the decision trees and the duration of each state is obtained from a duration model for given input text. Since pitch and spectral features are clustered separately by decision trees, the pitch and spectral state sequences are different. The whole training waveform corpus is already coded by tied state ID, as described in section 2.2. We use generated spectral tied-state ID sequence to search corpus. The matched waveform segments can be in length of one, two or multiple states, which depend upon the contextual matches between the state ID sequence in training sentence and that in a testing sentence. The corresponding waveform segments with different lengths are used to construct a candidate lattice.

Segment candidate lattice is first pruned by duration and then by the target score. Duration is used to prune the candidate segment whose duration is very different from the predicted duration. Target score is defined as:

$$D_{tar} = (N(d_p) + N(d_s) + N(d_d))/3 \quad (4)$$

where $N(d_p)$, $N(d_s)$ and $N(d_d)$ are normalized KLD score of pitch, spectrum and duration between target state and candidate state. A function is adopted to normalize the KLD score to 0~1. Lattice pruning is candidate length dependent, i.e. the segment candidates with different lengths are pruned individually according to a preset number of surviving hypotheses.

A compact lattice with non-uniform segment candidates is constructed after pruning. The best path is searched in the lattice by the score as:

$$D = wD_{tar} + D_{con} \tag{5}$$

where D_{tar} is the target score defined in Eq. 4, D_{con} is the concatenation score retrieved from concatenation score table, and w is the weight to compensate for the dynamic range difference between target and concatenation scores. In order to keep original continuity in waveform segment, we use non-uniform segment candidates to construct lattice. However, it will result in the best path searching in favor of candidates with longer durations. To balance this bias, we search the best path in any possible state boundary by dividing longer units into state-level units. Fig. 3 shows an example of the best path searching in a pruned lattice for given input text. The waveform segments in the best path are concatenated together at the optimal concatenation points by a triangular window, cross-faded in the time domain. The optimal concatenation points are obtained by maximizing the Normalized Cross-Correlation (NCC) between two windows located at the end of former segment and the start of latter segment.

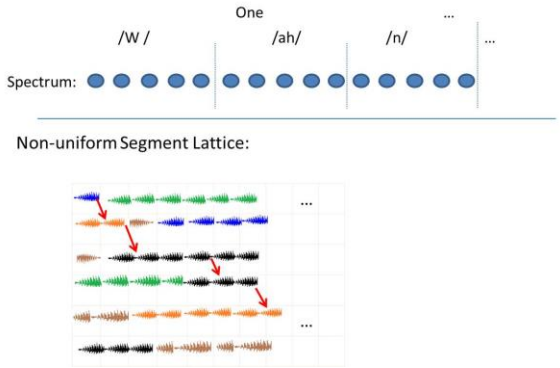

Fig. 3 An example of the best path searching in a pruned lattice for given input text. (Waveforms in different colors indicate non-contiguous waveform segments.)

2.4. Concatenation Score Table Construction

As mentioned in Section 2.3, we retrieve the concatenation score in a table instead of calculating it on the fly. Since NCC can not meet the positive, semi-definitive requirement of clustering, we use KLD between two normalized power spectra, s_p and s_q at the concatenation boundaries of segment p and segment q to measure concatenation discontinuity [3, 20]. It is defined as,

$$D_{KL}(s_p, s_q) = \frac{1}{M}\sum_{m=1}^{M}(s_p^m - s_q^m)\log\frac{s_p^m}{s_q^m} \tag{6}$$

We cluster the normalized power spectra of all possible concatenation segment boundaries by the criterion shown in Eq. 6, where the half of FFT power spectra (M=128 for 16ms frame length) is used. The Bregman divergence, in which KLD is included, is investigated as objective function for clustering and proved that it shows a monotonical non-increase of the objective function [22]. The algorithm of clustering is shown in Fig.4.

After clustering, we build up a KLD table for each cluster centroid pair. The KLD between two spectra s_p and s_q can be approximately evaluated by

$$D_{KL}(s_p, s_q) \approx \frac{1}{M}\sum_{k=1}^{K} d(C_i^{(k)}, C_j^{(k)}) \tag{7}$$

where $s_p \in C_i$ and $s_q \in C_j$, i, j, k are codebook and fragment indices.

1.	Initialize: S is spectrum set, s_n is the n-th spectrum. evenly divide the spectrum into K fragments; L_k indicates the length of k-th fragment, set a splitting threshold δ
2.	For k = 1 to K;
	2.1 $C(k)$ is the set of codes for k-th fragment; Initialize: $C(k) \leftarrow \emptyset$
	2.2 For n=1 to N If there doesn't exist $c \in C(k)$ such that $d(s_n^{(k)}, c) < \delta$, then $C(k) \leftarrow C(k) \cup \{s_n^{(k)}\}$
	2.3 Adjust each L_k and δ according to the size of codebook $\lvert C(k)\rvert$
3.	Assign each $s_n^{(k)}$ to the closest cluster center and do iteration till total clustering distortion is less than a threshold ρ or each $\lvert C(k)\rvert$ meets the requirement

Fig. 4 Fragment-based clustering for the normalized power spectra of all possible concatenation segment boundaries

3. EXPERIMENTS AND RESULTS

A phonetically and prosodically rich speech corpus, which consists of a female speaker's 9 hours recordings in Mandarin Chinese, is used in our experiments. Speech signals are sampled at 16kHz. The spectral analysis is performed by a 25-ms Hamming window and shifted every 5-ms. Spectral envelopes are estimated by STRAIGHT [16] and LPC analysis and ultimately represented by 40th order LSPs and their dynamic counterparts. F0 is extracted on a short-time basis by applying the robust algorithm for pitch tracking (RAPT) [17]. Five-state, left-to-right HMM phone models, where each state is modeled with a single Gaussian, diagonal covariance output distribution, are adopted. Rich phonetic and prosodic contexts are used as the question set in growing the decision trees. They include tones and breaks, quin-phone context, POS on contextual tri-word, positions of phone, syllable and word in phrase and sentence, and the length of syllable, word and phrase in number of phone, syllable and word. HMMs are firstly trained in ML sense and then refined in MGE sense. After HMM training, the whole training data are first force-aligned at the state level and then each state boundary is refined by the MGE criterion. The numbers of states for pitch, spectrum and duration are given in Table 1. The normalized power spectra of all possible concatenation segment boundaries are grouping into ~ 10,000 clusters. We set K=10 for total number of fragments. The dimensions of subvectors or fragments, which are automatically obtained in the clustering procedure, are {13, 10, 11, 11, 12, 13, 12, 12, 13, 21}.

Table 1. The numbers of states for pitch, spectrum and duration.

	Pitch	Spectrum	Duration
No. of State	14,328	7,393	1,249

50 sentences are used for developing set to tune the weight for target score in Eq. 5 and another 50 sentences are employed to test

the performance of our proposed method. The pruned lattice is constructed by the method mentioned in Section 2.3. The histogram of segment length in number of state in pruned lattice is shown in Fig. 5. 150 sentences synthesized by three systems are used for evaluation. The configurations of three systems are listed as following,

a) Baseline system: our previous HMM trajectory tiling based TTS [4].
b) New system A: it employs Gaussian kernel coded speech database, instead of generated trajectories from HMM in baseline system, to search the waveform segment candidates for constructing a lattice. The concatenation score in Eq. 5, which is used for best path search in lattice, is still NCC calculated in the real time.
c) New system B: it uses the concatenation score, which is retrieved from a table, together with the target score, in searching the best path in the lattice, which is the only difference from system A.

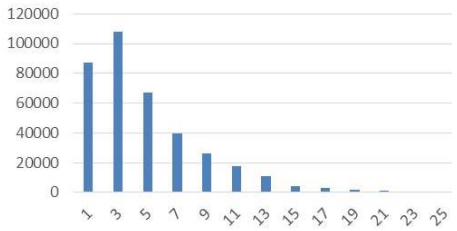

Fig. 5 Segment length in # of state in pruned lattice.

Both objective and subjective measures are used to evaluate the synthesis performance. Objective measures are target score for evaluating the distance from predicted state, segment length in number of state and average NCC per frame for measuring the smoothness or continuity of the concatenated waveform segment sequence. The subjective measure is an AB preference test between speech sentence pairs synthesized by two selected systems, i.e., between the baseline and the improved system. Ten native speakers participated in the subjective tests. The preference test demands a choice among a) former is better; b) latter is better; c) can't tell difference or can't tell which one is better; for each paired test stimuli in its naturalness and intelligibility. The order of stimuli presentation of the sentences is randomized.

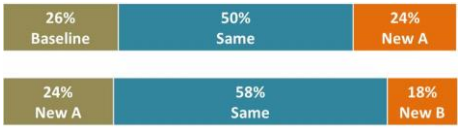

Fig. 6 The results of AB preference test for baseline and new proposed systems A and B.

The results of objective and subjective measures are shown in Table 2 and Fig. 6, respectively. The blank cells in Table 2 mean that the scores can be incomparable due to different criteria in baseline and new systems. The preference ratios are 26% vs. 24% and 24% vs. 18% for baseline and new system A and for system A and system B, respectively. It indicates that the new approach can essentially maintain the synthesis speech quality, in comparison with the baseline system. On the other hand, the computational complexity for the modules of the baseline and new systems A and B are listed in Table 3, where indicates the complexity of our new systems is significantly reduced.

Table 2. The results of objective measures for baseline and new systems A and B.

	F0	LSP	Duration	Seg Len	NCC
Baseline				6.9	0.98
New A	0.78	0.99	0.93	6.5	0.97
New B	0.82	0.99	0.92	6.2	0.95

Table 3. Computational complexities for the modules in our baseline and the two new systems A and B.

	Baseline	New A	New B
Trajectory Generation	O(TML)	None	none
Lattice Construction	O(TMS)	hash table lookup	hash table lookup
Best Path Search	$O(KH^2\ NlogN)$	$O(KH^2\ NlogN)$	$O(KH^2)$

* T: # of frames, M: feature dimension, L: delta feature window size, N: NCC window length, S: # of candidates per concatenation unit in corpus, K: # of concatenation points, H: # of candidates per concatenation unit in pruned lattice.

4. CONCLUSIONS

We propose a fast table lookup based, statistical model driven approach to non-uniform unit selection TTS. In training, the voice font of all waveform segments is structured as a Gaussian kernel coded hash table and a quantization table to pre-store concatenation costs between possible paired segments. In synthesis, the waveform segments with non-uniform length is first selected to construct a candidate lattice by looking up the Gaussian kernel coded hash table, and the best path is searched in the lattice by minimizing the accumulated concatenation scores, which are retrieved from the pre-computed quantization table. Experimental results show that the new approach can significantly reduce the search complexity without degrading synthesized TTS voice quality. (Demos for TTS: http://research.microsoft.com/en-us/projects/newfasttts/default.aspx)

5. RELATION TO PRIOR WORK

The work presented concentrates on how to construct a voice font as a search-efficient, Gaussian kernel coded hash table for all waveform segments and a quantization table for all possible pairs of concatenations. The approach significantly reduces the computational complexity in the backend of TTS speech synthesis for multi-channel, real time applications. The work by Ling and Wang [1] employs phone level unit selection by combing likelihood and KLD, Yan et al [2] uses rich-context (untied) model to represent the units parametrically and KLD between these models for unit selection, and Hirai et al [3] and Qian et al [4] used parameter trajectories to guide the unit selection process. While our present work proposes an approach to use tied state to quantize all waveform units and KLD between the predicted state and tied state for unit selection. In addition, quantized KLD lookup table of normalized power spectra is used to measure concatenation distortion instead of calculating them on the fly [3].

6. REFERENCES

[1] Z.-H. Ling and R.-H. Wang, "HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion", In Proc. ICASSP. pp. 1245–1248, 2007.

[2] Z.-J. Yan, Y. Qian and F.K. Soong, "Rich-context unit selection (RUS) approach to high quality TTS", In Proc. ICASSP, 2010.

[3] T. Hirai, J. Yamagishi and S. Tenpaku, "Utilization of an HMM-based feature generation module in 5 ms segment concatenative speech synthesis", In Proc. ISCA SSW6, 2007.

[4] Y. Qian, Z.-J. Yan, Y.-J. Wu, F. K. Soong, G.-L. Zhang and L.-J. Wang, "An HMM trajectory tiling (HTT) approach to high quality TTS – Microsoft entry to Blizzard Challenge 2010", in Proc. Blizzard Challenge Workshop, 2010.

[5] H. Zen, K. Tokuda and A. W. Black, "Statistical parametric speech synthesis", Speech Communication Volume 51, Issue 11, pp. 1039-1064, 2009.

[6] X . Gonzalvo, A. Gutkin, J. C. Socoró, I. Iriondo and P. Taylor, "Local minimum generation error criterion for hybrid HMM speech synthesis", In Proc. Interspeech, pp. 416-419, 2009.

[7] Z.-J. Yan, Y. Qian and F.K. Soong, "Rich context modeling for high quality HMM-Based TTS", In Proc. Interspeech, 2009.

[8] A. W. Black, H. Zen and K. Tokuda, "Statistical parametric speech synthesis", in Proc. ICASSP, pp. 1229-1232, 2007.

[9] T. Toda, and K. Tokuda, "Speech parameter generation algorithm considering global variance for HMM-based speech synthesis", in Proc. Interspeech 2005.

[10] Y. Qian, F.K. Soong and Z.-J. Yan, "A unified trajectory tiling approach to high quality speech rendering", IEEE Transactions on Audio, Speech and Language Processing, Issue 99, 2012.

[11] 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. Eurospeech, 1999.

[12] Y.-J. Wu and R.H. Wang, "Minimum generation error training for HMM-based speech synthesis", In Proc. ICASSP, 2006.

[13] K. Tokuda, T. Mausko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM", IEICE Trans. Inf. & Syst., vol.E85-D, no.3, pp.455-464, 2002.

[14] Y. Qian, H. Liang, F.K. Soong, "A Cross-Language state sharing and mapping approach to bilingual (Mandarin–English) TTS", *IEEE Transactions on Audio, Speech, and Language Processing,* VOL. 17, NO. 6, pp.1231-1239, 2009.

[15] K. Shinoda, and T. Watanable, "MDL-based context-dependent sub-word modeling for speech recognition", *J. Acoust. Soc. Jpn(E)*, vol.21, no.2, pp.79-86, 2000.

[16] H. Kawahara, I. Masuda-Katsuse and A. de Cheveigne, "Restructuring speech representations using pitch-adaptive time-frequency smoothing and instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds", Speech Communication, Vol.27, pp.187-207, 1999.

[17] A. D. Talkin, *Speech Coding and Synthesis, chapter A, Robust algorithm for pitch tracking (RAPT)*. Elservier Science B.V., Amsterdam, 1995.

[18] S. Kullback and R.A. Leibler, "On Information and sufficiency". Annals of Mathematical Statistics, Vol. 22, No.1, pp.79–86, 1951.

[19] P. Liu and F. K. Soong, "Kullback-Leibler divergence between two hidden Markov models", Microsoft Research Asia, Technical Report, 2005.

[20] Y. Stylianou and A. K. Syrdal, "Perceptual and objective detection of discontinuities in concatenative speech synthesis," in Proc. ICASSP, May 2001.

[21] http://synsig.org/index.php/Blizzard_challenge_2010.

[22] A. Banerjee, S. Merugu, I. S. Dhillon and J. Ghosh, "Clustering with Bregman divergences", Journal of Machine Learning Research, Vol. 6, pp.1705-1749, 2005.