A FULL TRAINING FRAMEWORK OF CROSS-STREAM DEPENDENCE MODELLING FOR HMM-BASED SINGING VOICE SYNTHESIS

Xin Wang[†]*, Minghui Dong*[‡]*, Zhen-Hua Ling*[†]

[†]National Engineering Laboratory of Speech and Language Information Processing
University of Science and Technology of China, Hefei, P.R.China
[‡]Human Language Technology Department, Institute for Infocomm Research, A*STAR, Singapore
wangx064@mail.ustc.edu.cn, mhdong@i2r.a-star.edu.sg, zhling@ustc.edu.cn

ABSTRACT

A cross-stream dependence modelling (CSDM) method has been proposed to model the dependence of spectral distributions on F0 observations for hidden Markov model (HMM) based speech synthesis. However, this method incorporates CSDM only for the embedded training of HMM estimation while ignoring CSDM in the clustering of context-dependent HMMs. This paper applies CSDM to HMM-based singing voice synthesis and presents a decision-tree-based model clustering method with explicit CSDM. This method, in conjunction with the previous CSDM method, forms a full CSDM training framework. Experimental results demonstrate that this full CSDM training framework achieves better performance than the previous CSDM method and the baseline without CSDM in a singing voice synthesis task.

Index Terms— hidden Markov model, singing voice synthesis, linear transform, model clustering

1. INTRODUCTION

The hidden Markov model (HMM)-based speech synthesis system (HTS) [1] is a popular solution to the lyrics-to-singing synthesis task [2, 3]. HTS simultaneously models the spectrum, F0 and other acoustic features with a unified HMM framework [1, 4]. After model training, it predicts the acoustic features [1] for input lyrics and music scores and then re-constructs the singing voice signal using a vocoder.

Although the synthesized singing voice is of good quality, it is still behind natural voice. One reason is the inconsistency between the nature of acoustic signals and the assumptions made in acoustic modelling. Typically, HTS assumes that the spectral and F0 features are independent from each other given HMM state sequences. However, researchers have observed the influence of F0 on vowel articulation [5]. Specifically in singing, the singers may consistently increase the frequency of the lowest resonance to match the pitch [6, 7]. Apart from this, the influence of F0 on the extraction of spectral features cannot be completely removed by vocoders [8]. Thus, the dependence between the two feature streams may not be negligible.

To address this issue for singing voice synthesis, a previously proposed cross-stream dependence modelling (CSDM) method for speech synthesis may be tried [8, 9]. Based on a group of linear transforms [10], this method incorporates observed F0 values into the mean vectors of spectral distributions at HMM states. After model training, this method can predict the spectral features in accord with the variation of the predicted F0 trajectory and thus increase the accuracy of the predicted spectral features.

However, this method only uses CSDM in the embedded training of model parameters while ignoring it in the decision-tree-based model clustering. Furthermore, the clusters of CSDM parameters are simply determined based on the decision-trees that are constructed without CSDM. To address this issue, this paper proposes a decision-tree-based model clustering method with explicit CSDM. Then, a full CSDM training framework for singing voice synthesis is presented by combining this model clustering method with the CSDM-based embedded training.

In the rest of the paper, Section 2 will describe the previous CSDM method and the proposed full CSDM training framework. Then, Section 3 will detail the experiments and the results. Finally, Section 4 will draw the conclusion.

2. CROSS-STREAM DEPENDENCE MODELLING

2.1. The conventional CSDM method

Existing systems based on HTS assume that acoustic features are generated by context-dependent HMMs. At time t, the acoustic feature vector $\boldsymbol{o}_t$ consists of a spectral part $\boldsymbol{c}_t$ and an F0 part $\boldsymbol{p}_t$ [2, 3]. For an HMM state $q_t = j$, wherein j denotes the index of a context-dependent state before model clustering, the probability to generate $\boldsymbol{o}_t$ is given by a state-dependent probability density function (PDF) $b_j(\boldsymbol{c}_t, \boldsymbol{p}_t)$. If $\boldsymbol{c}_t$ and $\boldsymbol{p}_t$ are assumed to be independent, $b_j(\boldsymbol{c}_t, \boldsymbol{p}_t)$ can be decomposed as $b_j(\boldsymbol{c}_t)b_j(\boldsymbol{p}_t)$.

To model the cross-stream dependence for singing voice

This work was partially conducted when Xin Wang was doing his research internship at Institute for Infocomm Research in 2015.

synthesis, our previous CSDM method [9] based on Continuous F0 HMM (CF-HMM) [11] can be utilized. It assumes that $b_j(\boldsymbol{c}_t, \boldsymbol{p}_t) = b_j(\boldsymbol{c}_t|\boldsymbol{f}_t)b_j(\boldsymbol{f}_t)b_j(l_t)$ wherein $\boldsymbol{p}_t$ consists of a continuous F0 feature $\boldsymbol{f}_t$ and a binary voiced/unvoiced flag l_t. The conditional PDF $b_j(\boldsymbol{c}_t|\boldsymbol{f}_t)$, which models the cross-stream dependence by a piecewise linear transform, is defined as

$$b_j(\boldsymbol{c}_t|\boldsymbol{f}_t) = \mathcal{N}(\boldsymbol{c}_t; \boldsymbol{W}_{r_j}\boldsymbol{\xi}_t + \boldsymbol{\mu}_{\boldsymbol{c},s_j}, \boldsymbol{\Sigma}_{\boldsymbol{c},s_j}), \quad (1)$$

wherein $\mathcal{N}(\cdot)$ is a Gaussian distribution; $\boldsymbol{\xi}_t = [\boldsymbol{f}_t^{\mathsf{T}}, 1]^{\mathsf{T}}$ is the extended F0 feature vector; $\boldsymbol{W}_{r_j} = [\boldsymbol{A}_{r_j}, \boldsymbol{b}_{r_j}]$ is the linear transform matrix for CSDM at state j and r_j denotes its cluster ID after model clustering; $\boldsymbol{\mu}_{\boldsymbol{c},s_j}$ and $\boldsymbol{\Sigma}_{\boldsymbol{c},s_j}$ are the residual mean and the covariance matrix at state j and s_j denotes their cluster ID. For neat expression, we define $\boldsymbol{\lambda}_{c,n} = \{\boldsymbol{\mu}_{\boldsymbol{c},n}, \boldsymbol{\Sigma}_{\boldsymbol{c},n}\}$, $\boldsymbol{\lambda}_c = \{\boldsymbol{\lambda}_{c,n}\}_{n=1,\dots,N}$, and $\boldsymbol{\lambda}_w = \{\boldsymbol{W}_m\}_{m=1,\dots,M}$, wherein N and M are the number of clusters for these two sets of model parameters.

The training process of our previous CSDM method [9] is shown on the left side of Fig.1. The conventional HTS training without CSDM is conducted at first, wherein context-dependent HMMs are clustered by decision trees [12] built under minimum description length (MDL) criterion [13]. Then, the decisions trees for the spectral stream are copied to cluster $\boldsymbol{\lambda}_c$ for the following CSDM training. Simultaneously, these decision trees are pruned for clustering $\boldsymbol{\lambda}_w$. Finally, $\boldsymbol{\lambda}_c$ and $\boldsymbol{\lambda}_w$ of all clusters are estimated by the embedded training with CSDM under maximum likelihood criterion [9].

2.2. The full CSDM training framework

Although our previous method [8, 9] incorporates CSDM in the embedded training, it exerts rigid constraints on the formation of $\boldsymbol{\lambda}_w$ clusters. For simplicity, we define $S(n) = \{j : s_j = n\}$ as the set of states sharing the same cluster $\boldsymbol{\lambda}_{c,n}$ and $R(m) = \{j : r_j = m\}$ as the states sharing $\boldsymbol{\lambda}_{w,m}$. Because the previous method directly uses the decision trees given by the conventional HTS training process to cluster $\boldsymbol{\lambda}_c$ while prunes these trees to cluster $\boldsymbol{\lambda}_w$, it requires that $S(n) \subseteq R(m)$ if $\exists i, s_i = n, r_i = m$. Furthermore, the parameters in $\boldsymbol{\lambda}_c$ are clustered before the embedded training with CSDM. The built decision trees may be incompatible with the model parameters after the embedded training with CSDM. Therefore, this paper presents a model clustering method with CSDM. In this method, the parameters in $\boldsymbol{\lambda}_w$ are clustered separately by building a decision tree under MDL criterion with CSDM rather than consulting the pruned decision tree of $\boldsymbol{\lambda}_c$.

Practically, to avoid the prohibitive computation cost in simultaneously growing the decision trees for $\boldsymbol{\lambda}_c$ and $\boldsymbol{\lambda}_w$, the proposed clustering process is decomposed into two iterative steps. First, the decision trees for clustering $\boldsymbol{\lambda}_w$ are built with $\boldsymbol{\lambda}_c$ and its decision trees fixed. Subsequently, the decision trees for clustering $\boldsymbol{\lambda}_c$ are re-built with $\boldsymbol{\lambda}_w$ and its decision trees fixed. Between the two clustering steps, the embedded training with CSDM is conducted to re-estimate $\boldsymbol{\lambda}_c$ and $\boldsymbol{\lambda}_t$.

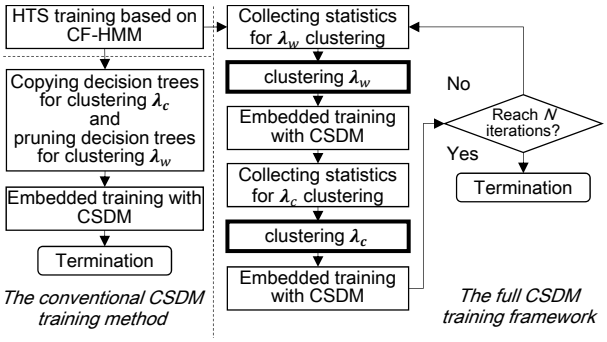


Fig. 1. Diagrams of the conventional CSDM training method (left) and the proposed full CSDM training framework (right).

As a result, an iterative full CSDM training framework can be formed as shown in Fig.1. Currently, the number of the full training iterations is manually set.

2.2.1. Clustering $\boldsymbol{\lambda}_w$ with CSDM

The MDL criterion [13] is followed to build state-position-dependent decision trees for clustering $\boldsymbol{\lambda}_w$ with CSDM. Here, *state position* refers to the position of a state in a phoneme HMM. Starting from a root note representing a global cluster of $\boldsymbol{\lambda}_w$, each leaf node splits into two nodes corresponding to the "yes" and "no" answers to a question on the contexts of HMMs. The MDL criterion is used to choose the optimal question for each split and decide when to stop splitting.

If the decision tree, U, has M leaf nodes, wherein each node represents a cluster of $\boldsymbol{\lambda}_w$, the description length of U is

$$\begin{aligned}\mathcal{D}(U) =& -\sum_{m=1}^{M}\sum_{\forall j, r_j=m}\sum_{t=1}^{T}\gamma_j(t)\log b_j(\boldsymbol{c}_t|\boldsymbol{f}_t) \\ &+ \frac{\alpha K M}{2}\log\Gamma + C \\ =& \frac{1}{2}\sum_{m=1}^{M}\big[\mathcal{V}(m) + \mathcal{L}(m)\big] + \frac{\alpha K M}{2}\log\Gamma + C,\end{aligned} \quad (2)$$

wherein K is the total number of parameters in $\boldsymbol{W}_m$, α is a hyper-parameter that controls the size of the built decision tree, C is a constant, and

$$\mathcal{V}(m) = \sum_{\forall j, r_j=m}\sum_{t=1}^{T}\gamma_j(t)\big[D\log(2\pi) + \log|\boldsymbol{\Sigma}_{\boldsymbol{c},s_j}|\big], \quad (3)$$

$$\begin{aligned}\mathcal{L}(m) = \sum_{\forall j, r_j=m}\sum_{t=1}^{T}&\gamma_j(t)(\boldsymbol{c}_t - \boldsymbol{W}_m\boldsymbol{\xi}_t - \boldsymbol{\mu}_{\boldsymbol{c},s_j})^{\mathsf{T}} \\ &\cdot \boldsymbol{\Sigma}_{\boldsymbol{c},s_j}^{-1}(\boldsymbol{c}_t - \boldsymbol{W}_m\boldsymbol{\xi}_t - \boldsymbol{\mu}_{\boldsymbol{c},s_j}),\end{aligned} \quad (4)$$

wherein D is the dimension of spectral feature vector $\boldsymbol{c}_t$. The state occupancy probability $\gamma_j(t)$ of frame t at state j is fixed during clustering [13], and $\Gamma = \sum_{m=1}^{M}\sum_{\forall j, r_j=m}\sum_{t=1}^{T}\gamma_j(t)$ is the number of training frames belonging to the M nodes.

Suppose one leaf node m in U splits into two nodes $m_{q,y}$ and $m_{q,n}$ in response to a question q, the change of

description length due to this split can be calculated as

$$\begin{aligned}\Delta_m(q) &= \mathcal{D}(U') - \mathcal{D}(U) \\ &= \frac{1}{2}\big[\mathcal{L}(m_{q,y}) + \mathcal{L}(m_{q,n}) - \mathcal{L}(m)\big] + \frac{\alpha K}{2}\log\Gamma.\end{aligned} \tag{5}$$

Because $\boldsymbol{\lambda}_c$ and its decision tree are fixed, $\mathcal{V}(m) = \mathcal{V}(m_{q,n}) + \mathcal{V}(m_{q,y})$ and $\Delta_m(q)$ does not rely on $\mathcal{V}$. Under the MDL criterion, the optimal question for node m is $q^* = \arg\min_q \Delta_m(q)$. However, if $\Delta_m(q^*) > 0$, the node m is not allowed to split. When all leaf nodes cannot split, the clustering process stops.

Evidently, the prerequisite for calculating $\Delta_m(q)$ is to estimate $\boldsymbol{W}_m$, $\boldsymbol{W}_{m_{q,y}}$, and $\boldsymbol{W}_{m_{q,n}}$ and then derive their corresponding $\mathcal{L}$ in (4). Taking $\boldsymbol{W}_m$ as an example, its estimation follows the formulae for the embedded training of CSDM [9]. Specifically, if $\boldsymbol{\Sigma}_{\boldsymbol{c},s_j}$ is diagonal, $\boldsymbol{W}_m$ can be estimated on a row-by-row basis, wherein the d-th row of the estimated $\hat{\boldsymbol{W}}_m$ is

$$\hat{\boldsymbol{w}}_m^{(d)} = \boldsymbol{k}_{m,d}^{\mathsf{T}} \boldsymbol{G}_{m,d}^{-1}, \tag{6}$$

wherein

$$\boldsymbol{G}_{m,d} = \sum_{\forall j, r_j = m} \sum_{t=1}^{T} \frac{\gamma_j(t)}{\sigma_{\boldsymbol{c},s_j}^{(d)}} \boldsymbol{\xi}_t \boldsymbol{\xi}_t^{\mathsf{T}}, \tag{7}$$

$$\boldsymbol{k}_{m,d} = \sum_{\forall j, r_j = m} \sum_{t=1}^{T} \frac{\gamma_j(t)}{\sigma_{\boldsymbol{c},s_j}^{(d)}} (c_t^{(d)} - \mu_{\boldsymbol{c},s_j}^{(d)}) \boldsymbol{\xi}_t, \tag{8}$$

and $c_t^{(d)}$, $\mu_{\boldsymbol{c},s_j}^{(d)}$ and $\sigma_{\boldsymbol{c},s_j}^{(d)}$ are the d-th element of $\boldsymbol{c}_t$, $\boldsymbol{\mu}_{\boldsymbol{c},s_j}$ and the diagonal vector of $\boldsymbol{\Sigma}_{\boldsymbol{c},s_j}$, respectively. Given $\hat{\boldsymbol{W}}_m$, $\mathcal{L}(m)$ can be calculated as

$$\mathcal{L}(m) = \sum_{d=1}^{D} (c_{m,d} - \boldsymbol{k}_{m,d}^{\mathsf{T}} \boldsymbol{G}_{m,d}^{-1} \boldsymbol{k}_{m,d}), \tag{9}$$

where

$$c_{m,d} = \sum_{\forall j, r_j = m} \sum_{t=1}^{T} \frac{\gamma_j(t)}{\sigma_{\boldsymbol{c},s_j}^{(d)}} (c_t^{(d)} - \mu_{\boldsymbol{c},s_j}^{(d)})^2. \tag{10}$$

Because $\gamma_i(t)$ is fixed, the statistics in (7), (8), and (10) can be efficiently accumulated based on the statistics collected in advance for the states belonging to the m-th cluster.

$\mathcal{L}(m_{q,y})$ and $\mathcal{L}(m_{q,n})$ can be calculated in the same way. Then, $\Delta_m(q)$ can be evaluated and the clustering process can be launched.

2.2.2. Clustering $\boldsymbol{\lambda}_c$ with CSDM

After clustering $\boldsymbol{\lambda}_w$, the parameters in $\boldsymbol{\lambda}_c$ and $\boldsymbol{\lambda}_w$ are re-estimated by embedded training with CSDM. Then, $\boldsymbol{\lambda}_c$ can be re-clustered under the same MDL criterion for clustering $\boldsymbol{\lambda}_w$. Because $\boldsymbol{\lambda}_w$ is fixed, it is easy to testify that to cluster the spectral distributions for $\boldsymbol{c}_t$ with CSDM into consideration is equivalent to directly clustering the spectral distributions for $\tilde{\boldsymbol{c}}_t = \boldsymbol{c}_t - \boldsymbol{W}_{r_j} \boldsymbol{\xi}_t$ that approximates the pitch-normalized spectral observation at frame t [14]. It is also possible to testify that, after clustering $\boldsymbol{\lambda}_w$, embedded training with CSDM, and collecting the state occupancy probabilities $\gamma_i(t)$, the conventional clustering method in HTS can be used to cluster the Gaussian distributions for $\tilde{\boldsymbol{c}}_t$, or parameters in $\boldsymbol{\lambda}_c$, as shown in Fig.1.

3. EXPERIMENTS

3.1. Corpus preparation

The database for experiments contains male solo singing recordings of Mandarin pop songs. The training set consists of 1000 randomly selected singing utterances and is about 70 minutes in length. The test and validation sets contain 100 utterances each. The spectral feature vector $\boldsymbol{c}_t$ consists line spectrum pairs (LSP) of order 40 derived from the spectral envelopes extracted by the STRAIGHT vocoder [15], an extra gain dimension and their delta and delta-delta coefficients. F0 was also extracted by STRAIGHT. Statistic show that 95% of the F0 data are in the range of 143-352 Hz (pitch note *D3* to *F4*). For the systems based on CSDM, these F0 values were interpolated into continuous trajectories [9].

3.2. System construction

Three types of experimental systems in Table 1 were constructed. All the systems used the initial/final of the Mandarin syllable as the acoustic unit. The contexts of acoustic units included phonemic, melodic and linguistic features, which were similar to those defined in [2]. Every context-dependent unit was implemented as a 5-state left-to-right HMM with Gaussian state distributions and diagonal covariance matrices. To reduce the number of free parameters, the matrix $\boldsymbol{A}_m$ in each $\boldsymbol{W}_m$ for *cCS* and *fCS* was defined as a three-block matrix [16] wherein each block models the dependency of the static, delta, or delta-delta spectral features on F0 features.

For *fCS*, the MDL factor α for clustering $\boldsymbol{\lambda}_c$ was 1. Meanwhile, α for clustering $\boldsymbol{\lambda}_w$ was tuned to 48 based on *fCS-1*'s likelihood on the validation set, which was similar to the method in [16]. Then, this α was fixed for *fCS-2* and *fCS-3*. Table 2 shows the cluster numbers of $\boldsymbol{\lambda}_w$ and $\boldsymbol{\lambda}_c$ for *fCS-n*. To ensure a fair comparison, the α for clustering $\boldsymbol{\lambda}_c$ in *BS* and *cCS* was tuned to 0.97 so that the number of $\boldsymbol{\lambda}_c$ clusters is comparable to that of *fCS*. The number of the $\boldsymbol{\lambda}_w$ clusters of *cCS* was tuned to a similar number as *fCS-3* by manually controlling the size of the pruned decision trees for $\boldsymbol{\lambda}_w$.

All the systems used the duration of the natural recordings for the objective and subjective evaluations. For *cCS* and *fCS*, *natural F0* of the test set and *synthetic F0* predicted by an independently built system with pitch-adaptive training [17] were used to predicted LSPs. The formulae to predict LSPs given models trained with CSDM can be found in [8].

Table 1. Experimental Systems.

ID	System description
BS	The baseline HTS system without using CSDM
cCS	The system using the conventional CSDM training method
fCS	The system using the full CSDM training method. *fCS-n* denotes *fCS* after the n-th full CSDM training iteration.

Table 2. Cluster numbers in $\boldsymbol{\lambda}_w$ and $\boldsymbol{\lambda}_c$ for different systems.

	BS	*cCS*	*fCS-1*	*fCS-2*	*fCS-3*
# clusters for $\boldsymbol{\lambda}_w$	-	54	30	39	54
# clusters for $\boldsymbol{\lambda}_c$	5169	5169	5016	5141	5161

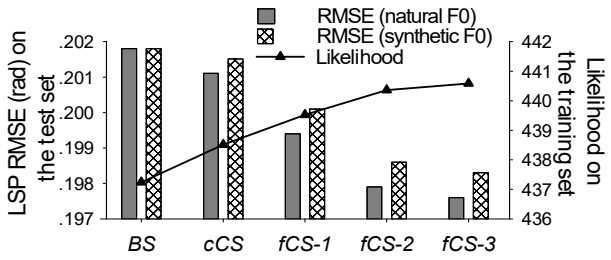


Fig. 2. The likelihoods on training set (line) and the LSP RMSEs on test set (bar) for different systems.

3.3. Objective evaluation

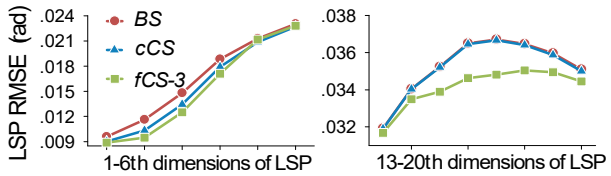
All the systems' likelihoods on the training set are shown by the line chart in Fig.2. Evidently, the likelihood of *fCS* rises consistently after each iteration of the full CSDM training. Fig.2 also shows the RMSEs of LSP prediction wherein the RMSE was calculated based on the method in [9]. A series of t-tests demonstrate that the RMSE of *BS* is significantly larger ($p < 0.05$) than the other systems using either natural or synthetic F0, except *cCS* using synthetic F0. Meanwhile, the RMSEs of *fCS-n* are significantly lower than those of *cCS*.

To inspect the objective results further, Fig.3 shows the RMSEs on the 1st to 6th and the 13th to 20th dimension of LSP. As Fig.3(a) demonstrates, both *cCS* and *fCS-3* can capture the cross-stream dependence for the 1st to 6th dimension of LSP. Furthermore, *fCS-3* can reduce the RMSEs of the 13th to 20th dimension of the predicted LSP. By examining the spectral envelopes and LSPs of some data frames, we find that the 13th to 20th dimension of LSP generally correspond to the second formant of the singing voice in our data. Fig.3(b) demonstrates that, when a synthetic F0 is used, *fCS-3* and *cCS*'s RMSEs on the 1st to 6th dimension of LSP increased while RMSEs on the 13th to 20th dimension stayed almost the same. One reason may be that the lower dimensions of LSP are sensitive to the small fluctuations of F0 in the singing voice such as overshooting and vibrato. Thus, the 1st to 6th dimension of the predicted LSP may be degraded by the over-smoothed synthetic F0.

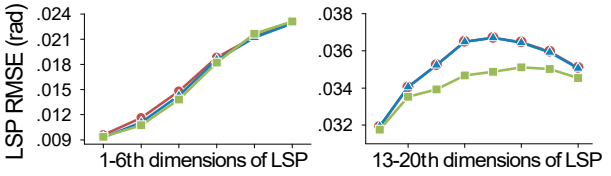
Because the RMSEs of *cCS* and *fCS-3* on the other dimensions of LSP were almost the same, they are not plotted in Fig.3. These dimensions of LSP may correspond to the parts of the spectrum that are insignificantly affected by F0.

3.4. Subjective evaluation

fCS-3 was compared with *cCS* and *BS* in subjective preference tests. In total, 12 native Mandarin speakers were invited to evaluate 20 randomly selected synthetic samples given by each system. The results are shown in Table 3. When natural F0 is used, *fCS-3* achieves better performance than *cCS*. However, the proportion of no preference is above 40%. One reason may be that, although *fCS* predicts LSPs more accurately, the increased accuracy may not lead to perceivable improvement on the quality of some synthetic samples. The comparison between *fCS* and *BS* shows similar



(a) LSPs predicted by *cCS* and *fCS-3* with natural F0.



(b) LSPs predicted by *cCS* and *fCS-3* with syntheitc F0.

Fig. 3. RMSEs of the 1-6th and 13-20th LSP dimensions on the test set. The RMSEs of BS are the same in (a) and (b).

results. Additionally, an informal test suggests that the *cCS* is not significantly different from *BS*, which is consistent with the our previous observations [8][9].

With synthetic F0, the performance of *fCS-3* drops while the proportion of no preference increases. The primary reason may be that the over-smoothed synthetic F0 degrades the accuracy of the lower dimensions of the predicted LSPs as Fig.3(b) demonstrates. Meanwhile, the synthetic F0 may make the synthetic samples out of tune and thus affects the perceived quality.

Nevertheless, all the results suggest that the proposed method models the cross-stream dependence better than the conventional CSDM method and the baseline method.

4. CONCLUSION

This paper presented a full CSDM training framework with a novel model clustering method incorporating CSDM. Experimental results demonstrated that the proposed framework achieved better performance than the conventional CSDM training framework. We also used this full CSDM framework to train a speech synthesizer for reading news. However, the improvement was insignificant. This may be due to the smaller F0 dynamic range of that speech corpus. For singing synthesis, the performance of CSDM could be further improved if a better F0 generation model such as the one leveraging lyrics information [18] could be incorporated. More generally, non-linear model may be tried in order to boost the power of CSDM. Besides, the soft-decision approach [19] for clustering CSDM may also be useful to ensure smooth change of CSDM parameters across adjacent HMM states.

Table 3. Results of subjective evaluation.

	BS	*cCS*	*fCS-3*	No pref.	p
Natural F0	-	20.0%	37.1%	42.9%	0.0004
	19.1%	-	37.9%	42.9%	<0.0001
Synthetic F0	-	22.5%	31.3%	46.2%	0.064
	22.1%	-	30.8%	47.1%	0.062

5. REFERENCES

[1] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[2] Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda, "An HMM-based singing voice synthesis system," in *INTERSPEECH-2006*, 2006, pp. 2274–2277.

[3] Keiichiro Oura, Ayami Mase, Tomohiko Yamada, Satoru Muto, Yoshihiko Nankaku, and Keiichi Tokuda, "Recent development of the HMM-based singing voice synthesis system-Sinsy," in *SSW7-2010*, 2010, pp. 211–216.

[4] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *EUROSPEECH-1999*, 1999, vol. 6, pp. 2347–2350.

[5] Kiyoshi Honda, "Relationship between pitch control and vowel articulation," *Vocal Fold Physiology Conference*, pp. 269–282, 1983.

[6] Elodie Joliveau, John Smith, and Joe Wolfe, "Acoustics: Tuning of vocal tract resonance by sopranos," *Nature*, vol. 427, no. 6970, pp. 116, Jan. 2004.

[7] Elodie Joliveau, John Smith, and Joe Wolfe, "Vocal tract resonances in singing: The soprano voice," *The Journal of the Acoustical Society of America*, vol. 116, no. 4, pp. 2434–2439, 2004.

[8] Zhen-Hua Ling, Wei Zhang, and Ren-Hua Wang, "Cross-stream dependency modeling for HMM-based speech synthesis," in *ISCSLP-2008*, 2008, pp. 1–4.

[9] Xin Wang, Zhen-Hua Ling, and Li-Rong Dai, "Cross-stream dependency modeling using continuous F0 model for HMM-based speech synthesis," in *ISCSLP-2012*. 2012, pp. 84–87, IEEE.

[10] Katsuhisa Fujinaga, Mitsuru Nakai, Hiroshi Shimodaira, and Shigeki Sagayama, "Multiple-regression hidden Markov model," in *ICASSP-2001*, 2001, vol. 1, pp. 513–516.

[11] Kai Yu and Steve Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE Transactions on Audio Speech & Language Processing*, vol. 19, no. 5, pp. 1071–1079, 2011.

[12] Julian James Odell, *The Use of Context in Large Vocabulary Speech Recognition*, Ph.D. thesis, University of Cambridge, 1995.

[13] Koichi Shinoda and Takao Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *Journal of Acoustic Society of Japan (E)*, vol. 21, no. 2, pp. 79–86, 2000.

[14] Harald Singer and Shigeki Sagayama, "Pitch dependent phone modelling for HMM based speech recognition," in *ICASSP-1992*, 1992, pp. 273–276.

[15] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveigne, "Restructuring speech representations using pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[16] Zhen-Hua Ling, Korin Richmond, Junichi Yamagishi, and Ren Hua Wang, "Integrating articulatory features into HMM-based parametric speech synthesis," *IEEE Transactions on Audio Speech & Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.

[17] Keiichiro Oura, Ayami Mase, Yoshihiko Nankaku, and Keiichi Tokuda, "Pitch adaptive training for HMM-based singing voice synthesis," in *ICASSP-2012*, 2012, pp. 5377–5380.

[18] S.W. Lee, Minghui Dong, and Haizhou Li, "A study of f0 modelling and generation with lyrics and shape characterization for singing voice synthesis," in *ISCSLP-2012*, 2012, pp. 150–154.

[19] Zhen-Hua Ling, Korin Richmond, and Junichi Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Transactions on Audio Speech & Language Processing*, vol. 21, no. 1, pp. 207–219, 2013.